Пояснювальна записка
до дипломного проекту
на тему: Підсистема балансування трафіка системи передачі данних з нестабільними каналами

Київ – 2019 рік

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

3G - (англ. 3rd Generation) — «третє покоління технології мобільного зв'язку» — набір послуг, який включає високошвидкісний мобільний доступ до мережі Інтернет та технологію радіозв'язку.

4G - (англ. 4th Generation) — стандарт четвертого покоління мобільного радіозв'язку, наступник 3G та 2G.

BPDU - (англ. Bridge Protocol Data Unit) — фрейм (одиниця даних) протоколу управління мережевими мостами, IEEE 802.1d, базується на реалізації протоколу STP (Spanning Tree Protocol). Використовується для виключення можливості виникнення петель в мережах передачі даних при наявності в них багатозв'язкової топології.

DHCP - (англ. Dynamic Host Configuration Protocol — протокол динамічної конфігурації вузла) — це стандартний протокол прикладного рівня, який дозволяє комп'ютерам автоматично отримувати IP-адресу та інші параметри, необхідні для роботи в мережі.

DSMLT – (англ. Distributed Split Multi-Link Trunking or Distributed SMLT) - технологія комп'ютерних мереж, розроблена компанією Avaya для вдосконалення протоколу розгалуження каналів (Multi Multi Link Trunking - SMLT).

EtherChannel — технологія агрегації каналів, розробка компанії Cisco Systems.

Ethernet - (англ. ether (ефір) та network (мережа)) — протокол кабельних комп'ютерних мереж, що працює на фізичному та канальному рівні мережевої моделі OSI.

Fast Ethernet - термін, що описує набір стандартів Ethernet для пакетної передачі даних з номінальною швидкістю 100 Мбіт/с, що в 10 разів швидше за початкову для Ethernet швидкість у 10 Мбіт/с.

Full-duplex mode - (англ. Full duplex mode) - повний дуплексний режим– функція прийомопередавача що дозволяє одночасно приймати і передавати сигнали.

Gigabit Ethernet - (GbE) — термін, що описує набір технологій для передачі пакетів Ethernet зі швидкістю 1 Гбіт/с. Він визначений в документі IEEE 802.3-2005.

GNU – (англ. «GNU's Not Unix») — вільна UNIX-подібна операційна система, що розробляється Проектом GNU

GSM - (англ. Global System for Mobile Communications, раніше фр. Groupe Spécial Mobile) - Глобальна система мобільного зв'язку— міжнародний стандарт для мобільного цифрового стільникового зв'язку з розділенням каналу за принципом TDMA та високим рівнем безпеки за рахунок шифрування з відкритим ключем.

Half-duplex mode – схема зв'язку, що дозволяє передавати сигнали в кожен момент часу тільки в одному напрямку.

HASH – функція, що використовувється для перевірки цілісності даних, їх ідентифікації та пошуку, а також дозволяє заміняти собою дані, які небезпечно зберігати в явному вигляді (наприклад, паролі, коди та інше).

HSR – (англ. High-availability Seamless Redundancy) - протокол паралельного резервування.

IGMP - (англ. Internet Group Management Protocol — протокол керування групами Інтернету) — протокол керування груповою (multicast) передачею даних в мережах, базованих на протоколі IP. IGMP використовується маршрутизаторами і IP-точками для об'єднання мережевих пристроїв в групи.

IP - реалізація ієрархічної схеми мережевої адресації.

ISP - (англ. Internet Service Provider) — провайдер послуг Інтернету.

LACP – (англ. Link Aggregation Control Protocol) - Протокол Управління Канальною Агрегацією для управління групуванням разом декількох фізичних портів для формування одного логічного каналу.

LAG - (англ. Link Aggrigation Group) - у термінології Vmware - логичний канал, об’єднуючий у собі uplink порти.

LEDE - (англ. Linux Embedded Development Environment) - проект, розпочатий частиною колишніх розробників OpenWrt.

Lua - (Луа, з порт. - «місяць») - скриптова мова програмування, розроблена в підрозділі Tecgraf (Computer Graphics Technology Group) Католицького університету Ріо-де-Жанейро (Бразилія). Інтерпретатор мови є вільно поширюваним, з відкритими вихідними текстами на мові Сі.

LuCI - (англ. Lua Configuration Interface) - інтерфейс конфігурації на Lua.

MAC (MAC-адреса) - (англ. Media Access Control) — управління доступом до носія) — це унікальний ідентифікатор, що зіставляється з різними типами устаткування для комп'ютерних мереж.

MM - багатомодове оптоволокно

MRC – (англ. Media Redundancy Clients) - протокол мережі передачі даних, стандартизованим Міжнародною електротехнічною комісією як IEC 62439-2. Специфікація протоколу дозволяє кільцям комутаторів Ethernet подолати будь-яку відмову з часом відновлення набагато швидше, ніж це досягається за допомогою протоколу STP.

MSTP - (англ. Multiple STP) - сучасна стандартна реалізація STP, що забезпечує як просте, так і повне підключення, призначене для будь-якої заданої віртуальної мережі (VLAN) по всій локальної мережі. MSTP використовує BPDU для обміну інформацією між сумісними пристроями, що з'єднують дерева, для запобігання циклів в кожному MSTI (декількох екземплярах Spanning Tree) і в CIST (Common і Internal Spanning Tree), вибравши активний і заблокований шляхи. Це робиться так само, як і в STP, без необхідності ручного включення резервних ліній і позбавлення від небезпеки мостових петель.

OpenWrt - операційна система для бездротових Wi-Fi маршрутизаторів, заснована на ядрі Linux.

Opkg – (англ. Open PacKage Management) - система керування пакетами на основі ipkg. Він написаний на C і має вигляд функції Advanced Packaging Tool (APT) / dpkg. Призначена для використання на вбудованих пристроях Linux і використовується в цій якості в проектах OpenEmbedded і OpenWrt.

PAgP - (англ. Port Aggregation Protocol) — протокол агрегування каналів; пропрієтарний протокол компанії Cisco Systems, що служить для автоматизації агрегування фізичних Ethernet-портів комутатора в один логічний.

Port Channel - технологія агрегації каналів, що була розроблена компанією Cisco Systems. Технологія дозволяє об'єднувати декілька фізичних каналів Ethernet в один логічний для збільшення пропускної здатності та підвищення надійності з'єднання.

PPPoE - (англ. Point-to-point protocol over Ethernet) - тунельний протокол, який дозволяє налаштовувати (або інкапсулювати) IP, або інші протоколи, які нашаровуються на PPP, через з'єднання Ethernet, але з програмними можливостями PPP-з'єднань.

PPTP - (англ. Point-to-Point Tunneling Protocol) — тунельний протокол типу точка-точка, що дозволяє комп'ютеру встановлювати захищене з'єднання з сервером за рахунок створення спеціального тунелю в стандартній, незахищеній мережі. PPTP поміщає (інкапсулює) кадри PPP в IP-пакети для передачі по глобальній IP-мережі, наприклад інтернет.

PRP - (англ. Parallel Redundancy Protocol) – протокол, описаний в стандарті IEC 62439-3.

PSTN - (англ. Public Switched Telephone Network, Телефонна мережа загального користування) — це планетарна телефонна мережа загального користування (ТМЗК), для доступу до якої використовуються звичайні проводові телефонні апарати, міні- АТС і обладнання передавання даних.

RSTP - (англ. Rapid STP) – наступна версія STP, що характеризується значними вдосконаленнями, серед яких зменшення часу збіжності і вища стійкість. Описаний в стандарті IEEE 802.1w (згодом включено до 802.1D-2004).

SMLT – (англ. Multi-link trunking) - протокол розділеного багатокагального

транкінгу, дозволяє кільком каналам Ethernet бути розділеним на кілька комутаторів.

STP - (англ. Spanning Tree Protocol, протокол кістякового дерева) — мережевий протокол, що працює на другому рівні моделі OSI.

TCP – (англ. Transmission Control Protocol, протоко́л керува́ння переда́чею) — разом із протоколом IP є стрижневим протоколом Інтернету, який дав назву моделі TCP/IP. Протокол призначений для управління передачею даних у комп'ютерних мережах, працює на транспортному рівні моделі OSI.

UCI – (англ. Unified Configuration Interface) - уніфікований інтерфейс розширеного налаштування.

UDP - (англ. User Datagram Protocol, протокол датаграм користувача) — один із протоколів в стеку TCP/IP. Від протоколу TCP він відрізняється тим, що працює без встановлення з'єднання. UDP — це один з найпростіших протоколів транспортного рівня моделі OSI, котрий виконує обмін повідомленнями (датаграмами — англ. datagram) без підтвердження та гарантії доставки.

VLAN - (англ. Virtual Local Area Network — віртуальна локальна комп'ютерна мережа) — є групою хостів з загальним набором вимог, що взаємодіють так, ніби вони прикріплені до одного домену, незалежно від їх фізичного розташування.

VOL – (англ. VPN Over LAG) – віртуальна приватна мережа через агреговані канали.

VPN - (англ. Virtual Private Network, віртуальна приватна мережа) — узагальнююча назва мереж, що створюються поверх інших мереж, які мають менший рівень довіри.

OpenvSwitch – програмна реалізація комутатора.

WAN – (англ. Wide Area Network) - комп'ютерна мережа, що охоплює величезні території (тобто будь-яка мережа, чиї комунікації поєднують цілі мегаполіси, області або навіть держави і містять у собі десятки, сотні а то і мільйони комп'ютерів).

X-Wrt - розширення OpenWrt для кінцевого користувача.

ВСТУП

У 1962 році академік А. А. Харкевич сформулював гіпотезу про те, що кількість інформації, яку треба збирати, обробляти і доставляти в потрібне місце, "зростає щонайменше пропорційно квадрату промислового потенціалу країни" [1]. Насправді, аналіз підтверджує, що в передових в технічному відношенні країнах таке зростання дійсно має місце приблизно зі ступенем 1,7-2,0. Це призводить до суттєвого зростання значущості діяльності, пов'язаної з виробництвом, передачею та переробкою інформації.

За даними ЮНЕСКО, в даний час більше половини працездатного населення розвинених країн прямо або побічно бере участь в процесі виробництва і розподілу інформації. Три провідні галузі інформаційного сектора суспільного виробництва (обчислювальна техніка, промислова електроніка і зв'язок) грають зараз для цих країн ту ж роль, яку на етапі їх індустріалізації грала важка промисловість.

Іншими словами, світова спільнота наближається до такої міри залежності свого існування від функціонування інформаційних мереж, яка порівнянна із залежністю від систем забезпечення електроенергією. Це крім очевидних переваг має і зворотну сторону. Відмова мережі зв'язку може мати наслідки, що перевершують наслідки аварій енергосистеми. У зв'язку з цим проблема оцінки і забезпечення надійності мереж є актуальною.

Надійність - це властивість об'єкта (системи), що полягає в його здатності виконувати задані функції за певних умов експлуатації. Кількісно надійність характеризується рядом показників, склад і спосіб визначення яких залежать від типу аналізованої системи.

Результати розгляду вимог до надійності системи повинні стати джерелом дій для побудови архітектури каналів зв’язку, що забезпечують необхідну пропускну спроможність та відмовостійкість. Використання просто швидкісного каналу зв’язку не завжди призводить до очікуваних результатів.

Безмежне використання захисних механізмів каналів зв’язку, таких, як механічні або електричні, часто вимагають значних фінансових витрат.

Таким чином постає завдання щодо створення у межах системи передачі даних підсистеми, що буде балансувати завантаження каналів зв’язку в залежності від їх пропускної спроможності, фізичного стану, термінів проходження даних. Реалізація цього завдання може бути вирішена шляхом організації надання пріоритетів на використання каналів.

Слід зазначити, що алгоритми балансування між каналами та їх програмна реалізація у розробників мережевого обладнання є досить захищеними об’єктами, оскільки оптимальний метод дозволить мати значні переваги перед конкурентами. Тобто конкретна реалізація балансування вже є елементом «ноу хау» в системі передачі даних.

Метою цього дипломного проекту є створення підсистеми балансування трафіку системи з нестабільними каналами. Завдання, яке поставлено для вирішення, є програмна реалізація балансування передачею даних між каналами зв’язку на основі аналізу їх стану. Перевірку стану каналів зв’язку розробляємо на основі процедур зворотнього зв’язку між вузлами обміну даними, що нададуть можливість аналізувати якісні показники каналів. Предметом дослідження є системи передачі данних на базі технології Ethernet.

Робота складається з вступу, огляду і аналізу систем балансування трафіку, опису технологій агрегування в операційних системах на базі linux, опису алгоритма та реалізації підсистеми балансування навантаження каналів зв’язку VOL з прикладами програмного забезпечення та протоколами її функціонування.

Практичне значення цього дипломного проекту полягає у можливості побудови апаратно-програмного комплексу надійного обміну даними у будь якій реалізації (сервер – сервер, сервер – користувач, сервер – накопичувачі інформації, та інше).

Апробація розробки була проведена шляхом побудови обміну даними між двома серверами, між якими були задіяні 2 канали зв’язку (Ethernet та 3G), та проведені випробування шляхом почергового відключення контрольованих

каналів. Розроблена система відреагувала на відключення каналів перенаправив трафік в справний канал, що дозволило продовжити обмін інформацією без суттєвого зниження пропускної спроможності.

Бакалаврський проект складається з наступних розділів: вступ, основні розділи, висновок, список використаних джерел із 20 найменувань, 3 додатків. графічна частина включає 4 креслеників формату А3. Загальний обсяг пояснювальної записки 68 сторінок та додатків 11 сторінок.

1 ОГЛЯД І АНАЛІЗ СИСТЕМ БАЛАНСУВАННЯ ТРАФІКУ

1.1 Види реалізації агрегування каналів

Резервування є універсальним заходом забезпечення надійності, що використовується у техниці. Резервування (або дублювання) каналів зв’язку підвищує пропускну спроможність мережі та надає можливість реагувати на проблеми з відмовами в окремих каналах зв’язку.

Використання різних каналів зв'язку дозволяє забезпечити багаторівневу систему резервування каналів передачі даних:

- Використання різнорідних каналів зв'язку: PSTN, GSM, 4G, Ethernet, тощо;
- Використання різних протоколів при організації обміну даними;
- Алгоритм перебору шляхів (каналів) обміну даних.

Кінцева мета дублювання каналів обміну даними здійснюється з метою виключення таких вузлів, відмова яких здатний вивести з ладу всю систему. Так, наприклад, якщо магазин підключений до центрального сервера за допомогою загальної системи передачі даних через єдиний канал зв’язку, то відмова цього каналу призведе до зупинки всіх касових та інших операцій і спричинить, відповідно, серйозні, в тому числі фінансові наслідки. Якщо ж даний канал дубльований іншим каналом, то відмова першого призводить до фактичного повернення мережі до первісного стану без резервування. Резервування дозволяє системі залишатися працездатною на період ремонту каналу, що вийшов з ладу.

Резервовані мережеві структури використовуються для двох цілей:

- регулювання навантаження на мережу - додавання резервного каналу зв'язку збільшує сумарну пропускну здатність початкового з'єднання, для цієї мети використовується агрегування каналів (LAG; IEEE 802.3ad);

- підвищення стійкості до збоїв - резервні канали зв'язку між вузлами мережі дозволяють системі перемикатися на запасні лінії зв'язку в разі відмови основної.

Незважаючи на те що друга мета може бути досягнута в рамках першої, пріоритет найчастіше віддається підвищенню стійкості до збоїв. У промислових мережах обсяги переданої інформації менше, ніж, наприклад, в офісних мережах, зате вимоги до надійності їх доставки вище. На гарантовану доставку даних в заданий проміжок часу орієнтовані промислові протоколи зв'язку і мережеві технології. Апаратні збої, звичайно, виключити не можна, але можна зробити їх наслідки найменш хворобливими.

У термінології комп'ютерних мереж балансування навантаження або вирівнювання навантаження трафіку – це розподіл завдань з передачі даних між декількома мережевими каналами зв’язку з метою оптимізації використання ресурсів, скорочення часу обміну інформації, а також забезпечення відмовостійкості (резервування). Один з методів досягнення вирівнювання навантаження – це використання агрегування каналів. Термін «Агрегування каналів» (в англійській мові link aggregation) - технологія об'єднання двох та більше паралельних каналів передачі даних в мережах Ethernet в один логічний, що дозволяє збільшити пропускну здатність і підвищити надійність [2]. У різних конкретних реалізаціях агрегування використовуються альтернативні назви:

- транкінг портів (Port trunking) в Cisco trunk'ом називається тегированний порт, тому з цим терміном плутанини найбільше);
- зв'язування каналів (Link Bundling);
- склейка адаптерів (NIC bonding);
- пару адаптерів (NIC teaming);
- Ethernet trunk в Cisco так називається агрегування каналів, це може ставитися як до налаштування статичних агрегованих каналів, так і з використанням протоколів LACP або PAgP;
- та безліч інших: Port Channel, Port Teaming, LAG (link aggregation), Multi-Link Trunking (MLT), DMLT, SMLT, DSMLT, R-SMLT, Network Fault Tolerance (NFT), Fast EtherChannel.

1.2 Протокол LACP

LACP (англ. Link aggregation control protocol) - відкритий стандартний протокол агрегування каналів, описаний в документах IEEE 802.3ad і IEEE 802.1aq. Багато виробників для своїх продуктів використовують не стандарт, а патентовані або закриті технології, наприклад, Cisco застосовує технологію EtherChannel (розроблену на початку 1990-х років компанією Kalpana), а також нестандартний протокол PAgP [3].

Link Aggregation Control Protocol (LACP) - протокол, призначений для об'єднання кількох фізичних каналів в один логічний в мережах Ethernet. Агреговані канали LACP використовуються як для підвищення пропускної здатності, так і підвищення відмовостійкості. Використання LACP в деяких випадках дозволяє виявити пошкоджений канал, який би при використанні звичайної статичної агрегації виявлений б не був. Описується стандартом IEEE 802.3ad [4].

Історія LACP починається з середини 1990-х років, коли більшість виробників мережевого устаткування включали технологію агрегування каналів в свої комутатори для збільшення їх пропускної спроможності. Однак кожна компанія розробляла власний протокол, що призводило до проблем з сумісністю. Тому в листопаді 1997 року на зустрічі ініціативної групи розробників IEEE 802.3 було вирішено створити інтероперабельності стандарт агрегації каналів. У нього також було вирішено включити функцію автоматичної конфігурації, за рахунок чого збільшувалася б і відмовостійкість. Стандарт став відомий як "Link Aggregation Control Protocol".

У березні 2000 року, після 2 років розробки, опис LACP було офіційно опубліковано як стандарт IEEE 802.3ad-2000 (стаття 43), названий так по імені робочої групи. Практично всі виробники мережевого обладнання швидко прийняли цей об'єднаний стандарт замість своїх фірмових розробок [5].

У 2006 році було піднято питання про перенесення LACP в групу стандартів

802.1, яка більш відповідала його становищу в стеку протоколів. Перенесення офіційно здійснилось 3 листопада 2008 року, коли стандарт був опублікований як 802.1AX-2008. Повна назва - 802.1AX-2008 IEEE Standard for Local and Metropolitan Area Networks - Link Aggregation (IEEE Стандарт для локальних і міських обчислювальних мереж - Агрегація каналів), але попередньє його розташування 802.3 ще досі використовується.

1.3 Агрегація на прикладі протоколу LACP

Відповідно до вимог протоколу пристрій, що ініціює обмін, відсилає пакети, які називаються LACPDU, через всі інтерфейсні пристрої, на яких він включений [6]. На підставі цих пакетів обладнання визначає приналежність фізичних портів до того чи іншого логічного каналу. Протокол може працювати в двох режимах:

- пасивний режим, при якому обладнання чекає від сусіда LACPDU пакети і тільки тоді починає висилати свої;
- активний режим, при якому обладнання постійно шле LACPDU пакети.

Для того, щоб LACP заробив, потрібно однакова швидкість і ємність каналів.

В результаті встановлення роботи протоколу LACP комутатори обмінюються:

- System Identifier (priority + MAC);
- Port Identifier (priority + номер порту);
- Operational Key (параметри порту).

1.4 Опис технології EtherChannel

EtherChannel - технологія агрегації каналів, розроблена компанією CiscoSystems. Технологія дозволяє об'єднувати декілька фізичних каналів Ethernet в один логічний для збільшення пропускної спроможності і підвищення надійності з'єднання [7].

EtherChannel дає можливість об'єднувати від двох до восьми портів Ethernet з швидкістю 100 Мбіт/с, 1 Гбіт/с або 10 Гбіт/с (всі порти в каналі повинні мати однакову швидкість), що працюють по кручений парі або по оптоволокну. Технологія дозволяє досягти результуючої швидкості до 80 Гбіт/с. Додатково, від одного до семи портів можуть бути неактивні і включатися в роботу при обриві з'єднання по одному з активних портів. При відсутності резервних портів, трафік автоматично розподіляється по всім активним з'єднанням.

Канал може встановлюватися між маршрутизаторами, комутаторами і мережевими адаптерами на сервері. Всі мережеві адаптери, які є частиною каналу, отримують один MAC-адресу, що робить канал прозорим для мережевих додатків. Балансування трафіку між портами провадиться на основі хеш-функції над MAC-адресою, IP-адресою або TCP і UDP портом джерела або одержувача. Таким чином, в деяких несприятливих випадках, весь трафік може передаватися по одній фізичній з'єднанню.

При використанні протоколу STP разом з EtherChannel, всі з'єднання в каналі розглядаються як одне логічне і BPDU надсилається тільки по одному з них. Спеціальний алгоритм дозволяє виявити невідповідності, коли один з комутаторів не налаштований для роботи з каналом.

При налаштуванні EtherChannel, порти на обох сторонах каналу додаються до нього вручну, або використовується один з протоколів автоматичної агрегації портів: пропріетарний протокол Cisco PAgP, або описаний в стандарті IEEE 802.3ad LACP [8].

Технологія EtherChannel дозволяє об'єднувати декілька фізичних портів на комутаторах в один логічний. Наприклад, два комутатора з'єднані між собою

чотирма проводами (на кожному використовується по 4 порта по 100 Mbit кожен). У разі відсутності EtherChannel - лінки ці будуть вважатися петлями і протокол Spanning Tree заблокує три з них. Працювати буде тільки один і швидкість обміну даними складе 100 Mbit, зате з потрійним резервуванням. У разі ж об'єднання їх за допомогою EtherChannel (на обох комутаторах), між комутаторами буде один віртуальний лінк, що складається з чотирьох фізичних, що працює зі швидкістю 400 Mbit.

Технологія EtherChanel спочатку була запропонована компанією Cisco, але зараз використовуються і у інших виробників. Технологія дозволяє кілька фізичних портів в один віртуальний порт, іменований PortChannel. Можливо створювати кілька окремих PortChannel-ів на одному комутаторі. Ця технологія має наступні переваги:

- головна - швидкість, немає необхідності купувати новий комутатор, якщо на старому тільки гігабітні порти, а нам треба більш швидкий аплинк до наступного комутатора, ми можемо об'єднати, наприклад, 8 портів і отримати 8 гігабіт;

- у плані надійності Etherchannel відрізняється від використання протоколу SpanningTree тим, що якщо в STP пропадає якийсь лінк, то починається перерахунок топології, що займає якийсь час, після чого, резервний канал вводиться в дію, в разі ж Etherchannel, топологія не змінюється, просто швидкість каналу в попередньому прикладі стане не 8, а 7 гігабіт. Іншими словами, EtherChannel не рятує від необхідності використовувати SpanningTree, але в разі, якщо лінк пропадає саме на агрегованому ділянці, позбавляє від необхідності перерахунку топології;

- ще одне невелика, але все ж перевага - після створення інтерфейсу portchannel, велика частина налаштувань може проводитися на ньому - немає необхідності окремо налаштовувати входять до нього фізичні інтерфейси - всі параметри будуть транслюватися на них автоматично.

Для того, щоб PortChannel міг існувати, необхідно, щоб всі вхідні в нього порти мали однакові параметри, а саме:

- однакову швидкість (не можна створити portchannel, куди входили б, наприклад, FastEthernet і GigabitEthernet, потрібно або всі 10 Мбіт, або всі 100, або все - гігабіт і так далі);

- однакові налаштування дуплексу;

- однакові налаштування VLAN-ів. У разі access портів - всі порти повинні бути в одному VLAN, в разі trunk портів - список дозволених VLAN (команда switchporttrunkallowedvlan) так само повинен збігатися.

Коли два комутатора «домовляються» один з одним про використання між ними агрегованого каналу, застосовується один з двох протоколів: PAgP або LACP. PAgP - розроблявся спочатку Cisco, а потім з'явився аналогічний відкритий стандарт LACP, який був оформлений у вигляді специфікації IEEE і використовується як на Каталіст, так і на комутаторах інших виробників. Іншими словами, в сучасних реаліях найкраще використовувати LACP, так як він сумісний з усіма, на відміну від PAgP. У функціональному ж плані протоколи аналогічні. PortChannel може бути налаштований в одному з трьох режимів: On, Active і Passive (в LACP) або On, Auto, Desirable (в PAgP відповідно). Для того, щоб між комутаторами піднявся і заробив portchannel, необхідно, щоб обидві сторони були налаштовані в режимі On, або одна була в режимі Active, а інша - Passive.

Так як для об'єднання в EtherChannel на інтерфейсах повинні збігатися багато настройки, простіше об'єднувати їх, коли вони налаштовані за замовчуванням. А потім налаштовувати логічний інтерфейс.

Перед об'єднанням інтерфейсів краще відключити їх. Це дозволить уникнути блокування інтерфейсів STP (або переведення їх в стан err - disable).

Для того щоб видалити настройки EtherChannel досить видалити логічний інтерфейс. Команди channel - group видаляться автоматично.

Створення EtherChannel для портів рівня 2 і портів рівня 3 відрізняється:

- для інтерфейсів 3го рівня вручну створюється логічний інтерфейс командою interface port-channel;

- для інтерфейсів 2го рівня логічний інтерфейс створюється динамічно;
- для обох типів інтерфейсів необхідно вручну призначати інтерфейс в EtherChannel. Для цього використовується команда channel-group в режимі настройки інтерфейсу. Ця команда пов'язує разом фізичні і логічні порти.

Після того як налаштований EtherChannel:

- зміни, які застосовуються до port-channel інтерфейсу, застосовуються до всіх фізичних портів, які присвоєні цьому port-channel інтерфейсу;
- зміни, які застосовуються до фізичного порту впливають тільки на порт на якому вони були зроблені.

1.5 Агрегування каналів NICTeaming

Технологія об'єднання мережевих адаптерів NICTeaming полягає в тому, що кілька фізичних адаптерів (NIC) об'єднуються в групу, в результаті чого виходить один єдиний логічний адаптер. Процес об'єднання називають teaming [9].

NICTeaming технологія не нова, але раніше її реалізація цілком залежала від виробників мережевого устаткування. Можливість об'єднувати мережеві адаптери в групу засобами операційної системи з'явилася тільки в WindowsServer 2012. NICTeaming в Server 2012 дозволяє об'єднувати в групу адаптери різних виробників, єдине обмеження - всі вони повинні працювати на одній швидкості. Об'єднати в NICTeaming можна до 32 мережевих адаптерів [10].

Частину функціоналу технології NICTeaming виконує протокол STP, що дозволяє використовувати кілька фізичних інтерфейсів для відмовостійкості. Але даний протокол не дозволяє здійснювати балансування трафіку і по суті в якийсь один момент часу використовує тільки один фізичний лінк для передачі даних. Хоча агрегірування дозволяє збільшити пропускну здатність каналу, але на практиці реальне балансування навантаження між усіма інтерфейсами в агрегованому каналі далека від ідеалу. Технології по балансуванню

навантаження в агрегованих каналах, як правило, орієнтовані на балансування за такими критеріями: MAC-адресами, IP-адресами, портів відправника або одержувача (за одним критерієм або їх комбінації). Агрегування каналів у багатьох виробників мережевого устаткування можливо тільки в режимі "точка-точка". Тобто можливо агрегувати канали починаються на одному пристрої і закінчуються на іншому пристрої.

Для того, щоб агрегування каналів запрацювало корректно потрібно виконати наступні вимоги:

- Максимальна кількість підтримуваних інтерфейсів в агрегованому каналі визначається виробником обладнання. (Найбільш часта цифра 4 або 8 інтерфейсів, в Vmware на поточний момент - 32);

- Всі фізичні інтерфейси в агрегованому каналі повинні мати однакові настройки швидкості і режиму full-duplex. (При цьому, LACP не підтримує half-duplex mode);

- Всі фізичні інтерфейси в агрегованому каналі повинні мати однакові настройки протоколу агрегування (LACP, статичний і ін.).

VMware підтримує тільки один канал EtherChannel на vSwitch або vNetwork Distributed Switch (vDS).

NIC Teaming по протоколу LACP в Vmware vsphere 5.5.x VMware ESXi 5.5.x, як і минулі версії підтримує роботу в режимі статичного тімінга. У такій реалізації користь LACP в порівнянні з Static Aggregation виявляється у наступному:

- Можливість використовувати Hot-Standby Ports. Якщо додати фізичних портів більше ніж підтримує обладнання (принаймні в cisco), тобто можливість використовувати зайві порти в якості портів hot-standby mode. Якщо станеться відмова активного порту, порт hot-standby автоматично замінить його. Цей пункт малоприйнятний до ESX, але тим не менше працездатний.

- Перевірка коректності конфігурації. агрегація каналів з використанням LACP не активується, якщо є якісь проблеми з конфігурацією. Це допомагає переконатися, що все налаштовано коректно. Статична агрегація

не робить будь-яких перевірок перед своїм задіянням, тобто потрібно заздалегідь бути впевненими, що все зроблено правильно.

- Failover (виявлення відмови). Якщо є dumb-пристрій між двома кінцями EtherChannel, наприклад media converter, і один з лінків, що йдуть через нього відмовляє, LACP це розуміє і перестає слати трафік в відмовив лінк. Static EtherChannel не моніторить стан лінків. Це не типова ситуація для більшості систем vSphere, але в ряді випадків це може виявитися корисним.

Архітектурно, LACP'ом в vSphere управляє модуль ядра, який взаємодіє з доменом lacp_uw. Ключова відмінність настройки статичної агрегації від LACP на vSphere Distributed Switch полягає в наступному: при використанні NIC teaming в статичної конфігурації на vSphere Distributed Switch створюються Uplink порти, яким зіставляли фізичні vmnic інтерфейси хостів. А вже Uplink інтерфейс використовували в якості вихідного інтерфейсу взаємодії із зовнішнім світом. А віртуальні машини, працюючи на будь-якому з хостів вибирали з асоційованого з Uplink фізичного інтерфейсу потрібний. При цьому, угруповання і балансування відбувалася на рівні створених Uplink інтерфейсів в налаштуваннях розподілених порт-груп в розділі "Teaming and failover". При налаштуванні Link Aggregation Control Protocol ми створюємо т.зв. LAG (Link Aggrigation Group) - деяка сутність, аналогічна Port Channel в Cisco - логічний інтерфейс, який об'єднує фізичні інтерфейси і на рівні якої використовуватимуться LACP (active-passive) і Load Balancing, до якої асоціюються фізичні інтерфейси одного хоста [11]. А вже до цього самого LAG приєднуються / асоціюються фізичні vmnic інтерфейси. І цей же LAG використовується в якості вихідного інтерфейсу.

Включені в агрегований канал порти називаються членами групи агрегування (LinkAggregationGroup). Кількість портів в групі залежить від моделі комутатора. У керованих комутаторах в групу можна об'єднати до 8 портів.

Один з портів в групі виступає в якості майстра-порту (master port). Так як

всі порти агрегированной групи повинні працювати в одному режимі, конфігурація майстра-порту поширюється на всі порти в групі. Таким чином, при конфігуруванні портів в групі досить налаштувати майстер-порт.

Важливим моментом при реалізації об'єднання портів в агрегований канал є розподіл трафіку по ним. Якщо пакети одного сеансу будуть передаватися за різними портам агрегованого каналу, то може виникнути проблема на більш високому рівні моделі OSI. Наприклад, якщо два або більше суміжних кадрів одного сеансу стануть передаватися через різні порти агрегованого каналу, то через неоднаковою довжини черг в їх буферах може виникнути ситуація, коли через нерівномірне затримки передачі кадру більш пізній кадр обжене свого попередника. Тому в більшості реалізацій механізмів агрегування використовуються методи статичного, а не динамічного розподілу кадрів по портах, тобто закріплення за певним портом агрегованого каналу потоку кадрів певного сеансу між двома вузлами. В цьому випадку всі кадри будуть проходити через одну і ту ж чергу і їх послідовність не зміниться. Зазвичай при статичному розподілі вибір порту для конкретного сеансу виконується на основі обраного алгоритму агрегування портів, тобто на підставі деяких ознак надходять пакетів. У комутаторах D-Link підтримується 9 алгоритмів агрегування портів:

- macsource - MAC-адреса джерела;
- macdestination - MAC-адреса призначення;
- mac_source_dest - MAC-адреса джерела і призначення;
- ip_source - IP-адреса джерела;
- ipdestination - IP-адреса призначення;
- ip_source_dest - IP-адреса джерела і призначення;
- l4_src_port - TCP / UDP-порт джерела;
- l4_dest_port - TCP / UDP-порт призначення;
- l4_src_dest_port - TCP / UDP-порт джерела і призначення.

У комутаторах D-Link за замовчуванням використовується алгоритм mac_source (MAC-адреса джерела).

На відміну від протоколу STP, всі надлишкові зв'язки в одному агрегированном каналі залишаються в робочому стані, а наявний трафік розподіляється між ними для досягнення балансування навантаження. У разі відмови однієї з ліній, що входять до такого логічного каналу, трафік розподіляється між рештою лініями.

В комутаторах MES алгоритм можна вибрати з наступних 4 видів балансування:

- src-dst-mac-ip - балансування заснована на MAC адресу джерела, MAC адресу призначення, IP адресу джерела і IP адресу призначення;

- src-dst-mac - режим за замовчуванням, балансування заснована на MAC адресу джерела, MAC адресу призначення;

- src-dst-ip - балансування заснована на IP адресу джерела і IP адресу призначення;

- src-dst-mac-ip-port - балансування заснована на MAC адресу джерела, MAC адресу призначення, IP адресу джерела і IP адресу призначення, на портах призначення TCP / UDP.

Алгоритм балансування вибирається командою: console (config) # Port-Channelload-balance.

Алгоритми роботи балансування src-dst-mac: sourceMAC (з 0 по 5 біт) операція XORdestinationMAC (з 0 по 5 біт) отримуємо HASH. Над HASH виконуємо операцію MOD x (x - кількість портів в LAG). Отримуємо Index порту в LAG.

Алгоритми роботи балансування src-dst-ip: IPsourceaddress (с 0 по 5 біт) операція XORIPsourceaddress (с 16 по 21 біт) операція XORIPdestinationaddress (з 0 по 5 біт) операція XORIPdestinationaddress (с 16 по 21 біт) отримуємо HASH. Над HASH виконуємо операцію MODx (x - кількість портів в LAG). Отримуємо Index порту в LAG.

Алгоритм роботи балансування src-dst-mac-ipIPsourceaddress (с 0 по 5 біт) операція XORIPsourceaddress (з 16 по 21 біт) операція XORIPdestinationaddress (з 0 по 5 біт) операція XORIPdestinationaddress (з 16 по 21 біт) операція

XORsourceMAC (з 0 по 5 біт) операція XORdestinationMAC (з 0 по 5 біт) отримуємо HASH. Над HASH виконуємо операцію MODX (x - кількість портів в LAG). Отримуємо Index порту в LAG.

1.6 Приклад налагодження агрегування каналів Cisco

Для агрегування каналів в Cisco може бути використаний один з трьох варіантів:

- LACP (Link Aggregation Control Protocol) стандартний протокол;
- PAgP (Port Aggregation Protocol) пропріетарний протокол Cisco;
- статичне агрегування без використання протоколів.

Так як LACP і PAgP вирішують одні й ті ж завдання (з невеликими відмінностями у можливостях), то краще використовувати стандартний протокол [12]. Фактично залишається вибір між LACP і статичним агрегируванням.

Статичне агрегування:

- переваги:

  - не вносить додаткову затримку при піднятті агрегованого каналу або зміні його налаштувань;

  - це варіант, який рекомендує використовувати Cisco.

- недоліки:

  - немає узгодження налаштувань з віддаленою стороною. Помилки в налаштуванні можуть привести до утворення петель;

Агрегування за допомогою LACP:

- переваги:

  - узгодження налаштувань з віддаленою стороною дозволяє уникнути помилок і петель в мережі;

  - підтримка standby-інтерфейсів дозволяє агрегувати до 16ти портів, 8 з яких будуть активними, а решта в режимі standby.

- недоліки:

- вносить додаткову затримку при піднятті агрегованого каналу або зміні його налаштувань;

- для застосування IEEE 802.3ad необхідний комутатор, що підтримує цю технологію.

Переваги застосування об'єднання каналів IEEE 802.3ad перед EtherChannel полягає в тому, що можна використовувати комутатори, які підтримують IEEE 802.3ad, але не підтримують EtherChannel. Крім того, IEEE 802.3ad забезпечує захист від збоїв адаптерів.

Після настройки об'єднання ліній IEEE 802.3ad сервер (система хоста) починає обмінюватися з суміжним комутатором блоками даних керуючого протоколу об'єднання ліній (LACPDU). Тільки активний канал, який може бути основним каналом або резервним адаптером, обмінюється LACPDU з суміжним комутатором.

Комутатор дозволяє об'єднати тільки ті адаптери, для яких встановлена однакова швидкість передачі даних (наприклад, 100 Мбіт / с або 1 Гбіт/с) і двобічний режим передачі. AIX допускає об'єднання адаптерів з різними швидкостями або режимами передачі даних, однак при об'єднанні таких адаптерів на комутаторі може виникнути помилка. Якщо комутатору не вдасться об'єднати адаптери, продуктивність мережі може значно знизитися.

Відповідно до специфікації IEEE 802.3ad все пакети з однаковим цільовим IP-адресою відправляються через один і той же адаптер. Отже, при роботі в режимі 802.3ad пакети завжди розподіляються стандартним, а не карусельним методом.

Об'єднання ліній IEEE 802.3ad підтримує функцію резервного адаптера, як і EtherChannel. Крім того, резервний адаптер підтримує LACP IEEE 802.3ad. Порт комутатора, підключений до резервного адаптера, також повинен підтримувати IEEE 802.3ad.

1.7 Приклад конфігурування на маршрутизаторах Cisco

Конфігурування маршрутизатора повинно проводитись у такій послідовності:

Створення групи портів:

QSW (config) # port-group 1

Додавання портів в певну групу з режиму конфігурації порту:

QSW (config-if-ethernet1 / 0/2) # port-group 1 mode?

active Transmit preference regardless of partner

on Add to port-group regardless any condition

passive Transmit preference unless partner is active

Для статичного об'єднання використовується mode on.

Для динамічного - active або passive.

LACP відсилає пакети, які називаються LACPDU, через все інтерфейси пристрої, на яких він включений [13]. На підставі цих пакетів обладнання визначає приналежність фізичних портів до того чи іншого логічного каналу. Протокол може працювати в двох режимах:

- пасивний режим, при якому обладнання чекає від сусіда LACPDU пакети і тільки тоді починає висилати свої;

- активний режим, при якому обладнання постійно шле LACPDU пакети.

Таким чином, для роботи LACP необхідно, щоб хоча б одна зі сторін, що бере участь в агрегації була в активному режимі, вона буде активним ініціатором LACP. Крім цього, потрібно однакова швидкість і ємність каналів.

Балансування трафіку здійснюється за допомогою вибору фізичного каналу відправником фрейма за допомогою обраного алгоритму. До основних і часто використовуваних можна віднести наступні алгоритми:

- по MAC-адресу відправника або MAC-адресу одержувача або з огляду на обидві адреси;

- по IP-адресі відправника або IP-адресою одержувача або з огляду на обидві адреси.

1.8 Технології та рішення для промислових мереж Ethernet

Одне з ключових вимог для промислових мереж Ethernet - відсутність петель або замкнутих маршрутів в топології, тобто між одержувачем і відправником кадру даних повинен бути єдиний шлях його доставки. Поява замкнутого маршруту в мережі викличе лавиноподібне зростання трафіку і, отже, перевантаження мережі, тому в традиційних мережах Ethernet уникають виникнення таких петель. У промислових мережах задача протоколів резервування - це моніторинг дубльованих каналів зв'язку з метою недопущення колізій і перерозподіл трафіку в аварійних ситуаціях. Протокол резервування повинен гарантувати логічне існування тільки одного шляху доставки повідомлення в конкретний момент часу при фізичному наявності декількох. З існуючих фізичних каналів зв'язку один вибирається основним, інші чекають в резерві.

Такий принцип був вперше застосований в протоколі STP, який відстежував стан каналів зв'язку і при виявленні обривів направляв трафік з відмовив каналу на резервний. Це означає, що зв'язок втрачається на деякий час, поки обрив виявиться, а новий канал передачі даних буде активований. Залежно від розмірів мережі і складності її топології час відновлення зв'язку може бути різним і заздалегідь його визначити не можна.

На основі протоколу STP можна сформулювати основні вимоги до протоколів резервування Ethernet в промисловому середовищі:

- певний час відновлення мережі: період часу від моменту розриву основного з'єднання до відновлення зв'язку з резервного з'єднанню повинен бути менше певної допустимої величини;

- вимоги до мережі: для протоколу повинні бути визначені допустима топологія мережі, максимальна кількість вузлів (комутаторів), типи з'єднань та ін.;

- протокол повинен базуватися на стандартизованому методі або алгоритмі. Тільки так можна гарантувати сумісність з мережевим устаткуванням та іншими мережевими протоколами.

Перша вимога традиційно для промислових мереж, що працюють в реальному часі. Протокол резервування може бути задіяний, якщо максимально можливий час відновлення задовольняє вимогам процесу або програми, для якого мережа передачі даних використовується.

1.8.1 Протокол RSTP / MSTP

В останні роки згаданий протокол STP був витіснений своїм більш швидким послідовником - протоколом RSTP (Rapid Spanning Tree Protocol), описаним у стандарті IEEE 802.1D-2004 [14]. Даний протокол підтримує безліч різних топологій і базується на тому, що з кожного мережного підключення виділяється деревоподібна структура, так що між будь-якими двома вузлами мережі в кожний момент часу існує єдиний маршрут передачі даних. Всі з'єднання, які не ввійшли в активний «дерево», вважаються резервними і не активні до зміни топології. Протокол базується на мостових з'єднаннях (BPDU - Bridge Protocol Data Units), серед яких вибирається кореневої міст (з'єднання комутатор-комутатор). Все інше «дерево» будується від нього. Будь-яка зміна в активному «дереві» означає зміну в складі BPDU з розсилкою спеціального BPDU-кадру по вузлах мережі, після чого ті активують резервні маршрути для трафіку. В протокол вбудовано захист від перевантаження мережі даними кадрами, яка може привести до збільшення часу відновлення.

Протокол RSTP підтримує велику кількість вузлів і забезпечує час відновлення зв'язку близько 1 секунди. Це час багато в чому залежить від місця виникнення обриву зв'язку в мережі, тому не може бути чітко визначено заздалегідь.

Даний істотний недолік можна обійти, якщо обмежити топологію мережі

кільцевою структурою. Тим самим можна домогтися дотримання часу відновлення мережі близько 100 мс і менше.

MSTP - нова ітерація описаного протоколу резервованих «дерев», і працює вона за тим же принципом. Якщо RSTP працює незалежно від віртуальних мереж VLAN, то структури MSTP, навпаки, існують в складі віртуальної мережі і таким чином забезпечують більше зручностей і свобод в плані можливих топологій і розподілу навантаження. Обидва протоколу сумісні між собою і можуть бути реалізовані всередині однієї мережі.

Кільцева топологія зручна насамперед тим, що з нею досягається певний і гарантований час відновлення зв'язку після збою, що враховує кількість комутаторів в кільці. Стандарт IEC 62439-1 описує приклад розрахунку часу відновлення для кільця, а також додаткові обмеження (наприклад, RSTP не може бути налаштований на портах комутатора, не задіяних в кільці). Однак RSTP ні розроблений для кільцевого резервування, тому поступається спеціалізованим протоколам типу MRP. У комутаторах Hirschmann реалізована підтримка обох протоколів, і, хоча приписів щодо їх спільного застосування немає, використовувати в цьому випадку краще MRP.

1.8.2 MRP - стандартизоване резервоване кільце

Протокол MRP був спеціально розроблений для промислового застосування. Він описаний в стандарті IEC 62439-2 для промислових мереж Ethernet з високим ступенем доступності. MRP підтримує тільки кільцеву топологію мережі з кількістю комутаторів не більше 50, гарантуючи заздалегідь певний час відновлення зв'язку в разі виникнення збою. Час відновлення залежить від обраних параметрів протоколу MRP і може складати від 10 до 500 мс, причому максимальне час можна встановити заздалегідь. Наприклад, при максимальному часу відновлення, що дорівнює 200 мс, типове значення складе 50-60 мс при середньому завантаженні мережі.

Протокол передбачає об'єднання в кільце групи комутаторів, один з яких бере на себе роль ведучого (MRM - Media Redundancy Manager). Він контролює цілісність кільця, передаючи по кільцю тестові кадри даних в одну сторону і отримуючи їх по ланцюжку з іншого боку. Для запобігання колізій всі дані, крім тестових кадрів, блокуються на одному з двох кільцевих портів MRM-комутатора, утворюючи фактично лінійну топологію мережі. Якщо ведучий комутатор не отримує тестові кадри, це означає розрив кільця, в такому випадку він розблокує друге з'єднання, відновивши передачу даних.

Решта комутатори в кільці грають роль ведених (MRC - Media Redundancy Clients) і передають тестові кадри по ланцюжку з одного кільцевого порту в інший. Також відомі комутатори передають ведучому інформацію про зміну стану їх портів. Якщо MRM-комутатор отримав повідомлення від MRC-комутатора про відмову його кільцевого порту раніше, ніж недорахувався тестових кадрів, то він керується цим попередженням і активує заблоковане з'єднання. Такий підхід забезпечує найменше можливий час відновлення мережі.

1.8.3 PRP - паралельне резервування

Незважаючи на швидкість роботу MRP і його універсальність для широкого кола завдань, існують додатки, де неприпустимо навіть мінімальне час відновлення мережі. Для таких додатків необхідний зовсім новий підхід до питання високої доступності мережі. В основі цього підходу - існування мінімум двох одночасно активних сполук між двома вузлами мережі таким чином, що відправник інформації посилає кадри даних синхронно з двох Ethernet- каналах. Одержувач ж за допомогою протоколу резервування приймає перший кадр даних і відхиляє другий. Якщо другий кадр даних не отримана, адресат робить висновок про обрив зв'язку у відповідному каналі.

Даний механізм резервування реалізований в протоколі PRP (Parallel Redundancy Protocol), описаному в стандарті IEC 62439-3 [15]. PRP використовує дві паралельні мережі передачі даних з довільною топологією, не обмеженої ні

кільцями, ні іншими структурами. Більш того, в двох паралельних мережах може не бути резервування зовсім, а можуть застосовуватися протоколи MRP і RSTP. Таким чином, принципова перевага PRP полягає в його «безшовному» резервування з відсутністю навіть малого часу перемикання з основного на резервний канал зв'язку. Високий рівень доступності мережі з паралельним резервуванням дотримується за умови, що обидві підмережі, об'єднані PRP, не можуть відмовити одночасно.

Протокол PRP реалізується на кінцевих пристроях. Комутатори мережі працюють незалежно від даного протоколу і, відповідно, не повинні мати ніякої спеціальної апаратної або програмної підтримкою.

Кінцеві пристрої з підтримкою PRP (DANP -Double AttachedNode for PRP) мають два мережевих інтерфейсу і підключаються до двох незалежних мереж. При цьому мережі можуть мати різну топологію, середу і швидкість передачі. До мережі можуть підключатися і звичайні кінцеві пристрої з одним мережевим інтерфейсом (SAN - Single Attached Node). Також можуть використовуватися кінцеві пристрої типу DANP в ролі проксі-серверів (так звані RedBox - скорочення від Redundancy Box), до яких підключені декілька SAN-пристроїв. Від SAN- пристрою не потрібно ніякої спеціальної підтримки PRP. Цю можливість зручно застосовувати на практиці, користуючись тим, що в мережах з високою доступністю наявність паралельного резервування критично не для всіх пристроїв, тому кінцеві пристрої за ступенем важливості можна розділити на типи DANP і SAN і з'єднати, використовуючи дубльований або єдиний канал зв'язку відповідно. Кінцеві пристрої з можливістю паралельного резервування типу DANP повинні контролювати дубльовані кадри Ethernet. Отримавши дані для передачі в мережу, пристрій, що реалізує протокол PRP, посилає їх по двом мережних інтерфейсів одночасно. Таким чином, два кадри Ethernet відправляються з незалежних мереж до одного одержувачу і, з огляду на різну топологію і пропускну здатність обох мереж, доходять до адресата з різною затримкою. Перший прийшов одержувачу кадр приймається і передається на верхній рівень, другий - видаляється. В результаті мережеве додаток, що

використовує отримані дані, не "відчуває" різниці між резервованим з PRP і звичайним Ethernet-інтерфейсом.

Ідентифікація дублюючих кадрів здійснюється за спеціальним контрольному маркера - RCT (Redundancy Control Trailer), поміщений в Ethernet-кадр PRP-пристроєм. На додаток до ідентифікатора підмережі і призначених для користувача даних в кадр поміщається 32-бітове поле, що включає номер послідовності PRP. За цим номером кінцеве пристрій ідентифікує кадр і або передає його на верхній рівень, або видаляє. RCT-маркер знаходиться в кінці блоку даних, тому такий формат Ethernet- даних зчитується як DANP-, так і SAN-пристроями. Це властивість дозволяє мережевим пристроям обмінюватися інформацією за відсутності резервування.

В цілому протокол PRP дозволяє створити мережу з високою степом доступності, довільною топологією, але вимагає значно більших витрат на обладнання, інфраструктуру і мережеві компоненти.

1.8.4 HSR - безшовне резервування

Протокол HSR (High-availability Seamless Redundancy) - подальший розвиток ідеї паралельного резервування. Однак якщо у випадку з PRP йшлося про резервування мережі, то HSR - це протокол резервування з'єднань. HSR, як і PRP, описаний в стандарті IEC 62439-3. Але на відміну від PRP протокол HSR розроблений для кільцевої топології мережі. Як і PRP, він використовує два мережевих порту у кінцевого пристрою для підключення до мережі, але ланцюгом, замкнутого в кільце [16].

Формат кадру даних у протоколу HSR аналогічний PRP. Ідентифікатор HSR схожий на поле RCT: включає розмір призначених для користувача даних, тип порту відправника (1-й або 2-й порт) і номер послідовності. Однак якщо ідентифікатор протоколу PRP йде всередині стандартного Ethernet-кадру, то у випадку з HSR ідентифікатор протоколу йде на початку. Тому HSR-пристрої

розпізнають дані на льоту і швидше їх обробляють, передаючи з першого на другий інтерфейс по ланцюгу. При цьому кожне кінцеве пристрій пропускає через себе всі кадри даних, читає заголовки і відбирає собі кадри зі своєю адресою одержувача, а також широкомовні повідомлення. Для запобігання циркуляції по колу широкомовних повідомлень пристрій-відправник видаляє повідомлення, які пройшли повне коло по мережі.

На відміну від мережі з паралельним резервуванням, в HSR-кільце не можна включити стандартний пристрій з одним мережевим інтерфейсом - кільце не буде замкнуто і формат даних з HSR-заголовком не розпізнає. Аналіз кадру даних на другому рівні OSI з ідентифікатором PRP (він знаходиться в полі додаткової інформації) можливий і звичайним пристроєм - воно просто пропустить поле з RCT.Формат даних cHSR-заголовком виходить нестандартний, і кінцеве пристрій без підтримки HSR-протоколу його не розпізнає. Проте, в цьому випадку можна використовувати посередника RedBox, який включається в HSR-кільце і має додаткові підключення до кінцевих пристроїв поза кільця.

Як з'ясовуэться, стандартні пристрої «не розуміють» HSR-дані, проте самі HSR-пристрої «розуміють» стандартний формат даних. Це необхідно для конфігурації і діагностики вузлів кільця. При цьому стандартні кадри даних не проходять по колу, як HSR-дані, а пересилаються безпосередньо між станцією управління і пристроєм.

HSR-кільце починає роботу в штат- ном режимі тільки після відключення станції управління і замикання ланцюга. HSR-кільця можна з'єднувати між собою двома 4-портовими пристроями, званими QuadBox. Пристрої дублюють один одного, тому загальна мережа також залишається резервованої.

Що стосується часу відновлення, то тут HSR-протокол поводиться аналогічно PRP: кадри даних одночасно розсилаються по двох портів в обох напрямках по кільцю, в разі збою одна з черг даних досягне одержувача. Такий підхід гарантує резервування з нульовим часом відновлення і в той же час не вимагає додаткової мережевої структури.

З недоліків HSR можна відзначити обмежену гнучкість (тільки кільцева топологія), дворазовий обсяг трафіку, що передається по мережі з дублюванням кадрів даних, складність реалізації (спеціальний FPGA-чіп в кожному пристрої, синхронізація по протоколу IEEE 1588).

1.8.5 Висновок для вибору архітектури побудови промислових мереж

На практиці не існує ні ідеальної мережевої топології, ні ідеального протоколу резервування, що задовольняє всім вимогам промислових мереж. Правильний вибір топології мережі і протоколу резервування залежить від багатьох факторів, таких як фізичні вимоги до розташування мережевих компонентів.

Протокол HSR є новим (стандарт IEC 62439-3 прийнятий в лютому 2010 року) і перспективним. Серед основних сфер його застосування слід зазначити АСУ в енергетиці. Він навіть буде включений в другу версію стандарту для електричних підстанцій IEC 61850. Протокол HSR буде забезпечувати функціонування мережі Ethernet в реальному часі разом з протоколом синхронізації годин IEEE 1 588.

Для підвищення надійності і гнучкості мережі протоколи резервування можна комбінувати між собою. Можна зробити прогноз, що в майбутньому протоколи PRP і HSR (їх подальші ітерації) витіснять існуючі протоколи кільцевого і паралельного резервування.

Впровадженням протоколів PRP iHSR в реальні рішення займаються провідні світові розробники мережевого обладнання, такі як Siemens, Hirschmann (Belden), ZHAW. Також над ідеями паралельного резервування працюють компанії CISCO, RuggedCom. Перші мікросхеми FPGA виробництва компаній Altera і Xilinx з реалізацією цих протоколів існують з середини 2010 року. Механізми протоколів HSR і PRP були успішно протестовані з емуляцією на програмному рівні. Про функціонування в реальному часі з програмною

реалізацією, звичайно, мова не йде, зате можна позитивно оцінити їх працездатність у великих мережах, взаємодія з іншими протоколами резервування другого рівня OSI, GOOSE-повідомленнями протоколу IEC / IEC 61850.

1.9 Реалізація балансування від D-Link

У термінології комп'ютерних мереж балансування (вирівнювання) навантаження - розподіл передачі пакетів між декількома каналами зв’язку оптимізації використання ресурсів і скорочення часу сумарної передачі інформації.

Зазвичай системи балансування завантаження каналів використовують можливості рівня L4 (UDP / TCP). При цьому контролюється доступність кінцевого пристрояза IP-адресою і номером порту і приймається рішення: яким з доступних каналів слід переслати пакет. Найбільш часто для вибору каналу використовується карусельний алгоритм (round-robin). У більш просунутих варіантах алгоритму використовується аналіз рівеняякості каналу, проявом якої є перевірка його швидкості та перешкодозахищеності.

Для досягнення максимальної пропускної спроможності і відмовостійкості мережеві канали дозволяють розподілити або збалансувати навантаження, використовуючи всі наявні канали зв’язку одночасно. Наприклад, можна уникнути такої ситуації, коли в підключення до Internetчерез двох провайдерів по мережі пакети йдуть через одного провайдера, в той час як вихід в Інтернет через іншого простоює без навантаження. Або розподілити сервіси і направити трафік через всі наявні Інтернет-канали. Можливе налаштування балансування навантаження, якщо з'єднання з провайдерами здійснюються з різними типами підключення (Static IP, PPPoE, PPTP / L2TP), а також - для балансування трафіку, що проходить через VPN-тунелі, встановлені на різних фізичних інтерфейсах.

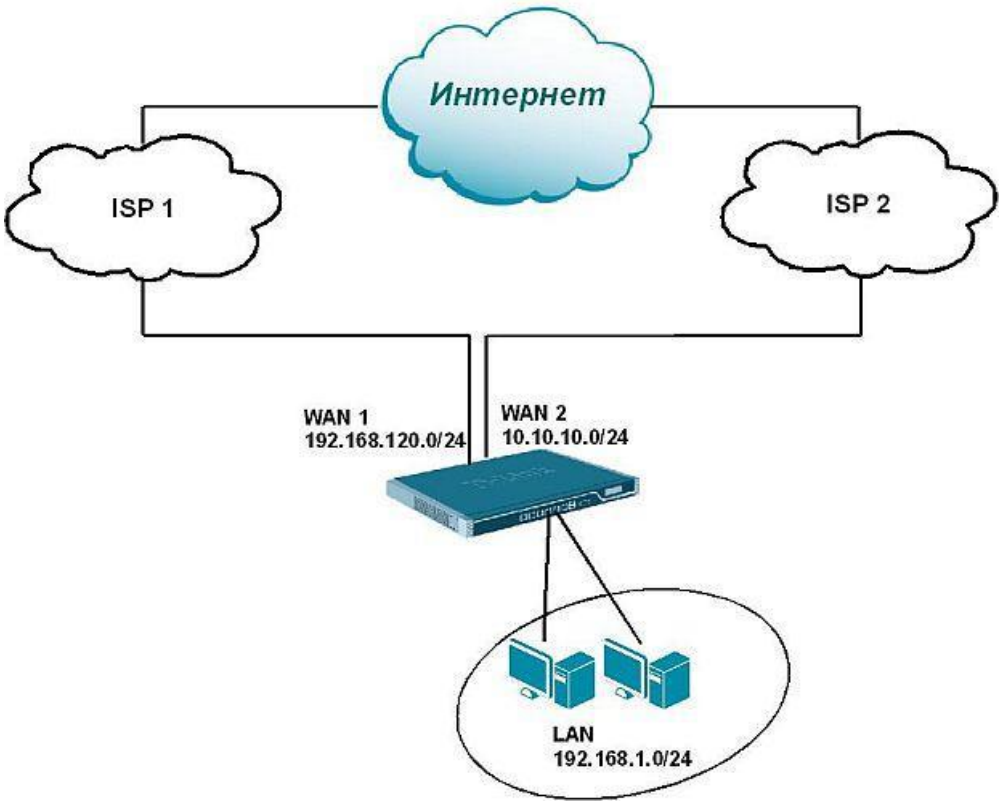

Рисунок 1.1 - Балансування навантаження між 2 фізичними каналами доступу

У міжмережевих екранах D-Link серії NetDefend передбачена функція, призначена для балансування мережного навантаження по різних маршрутах - RouteLoadBalancing (RLB), можливості якої забезпечують [17]:

- балансування трафіку між інтерфейсами на основі політик;
- балансування навантаження трафіку при одночасному множині доступі в Інтернет, користуючись послугами двох і більше провайдерів;
- балансування трафіку, що проходить через VPN-тунелі, встановлені на різних фізичних інтерфейсах.

Функція балансування навантаження в міжмережевих екранах NetDefend активується на основі таблиці маршрутизації шляхом створення об'єкта RLBInstance, в якому визначені два параметра: таблиця маршрутизації і RLB-алгоритм. З таблицею маршрутизації може бути пов'язаний тільки один об'єкт Instance.

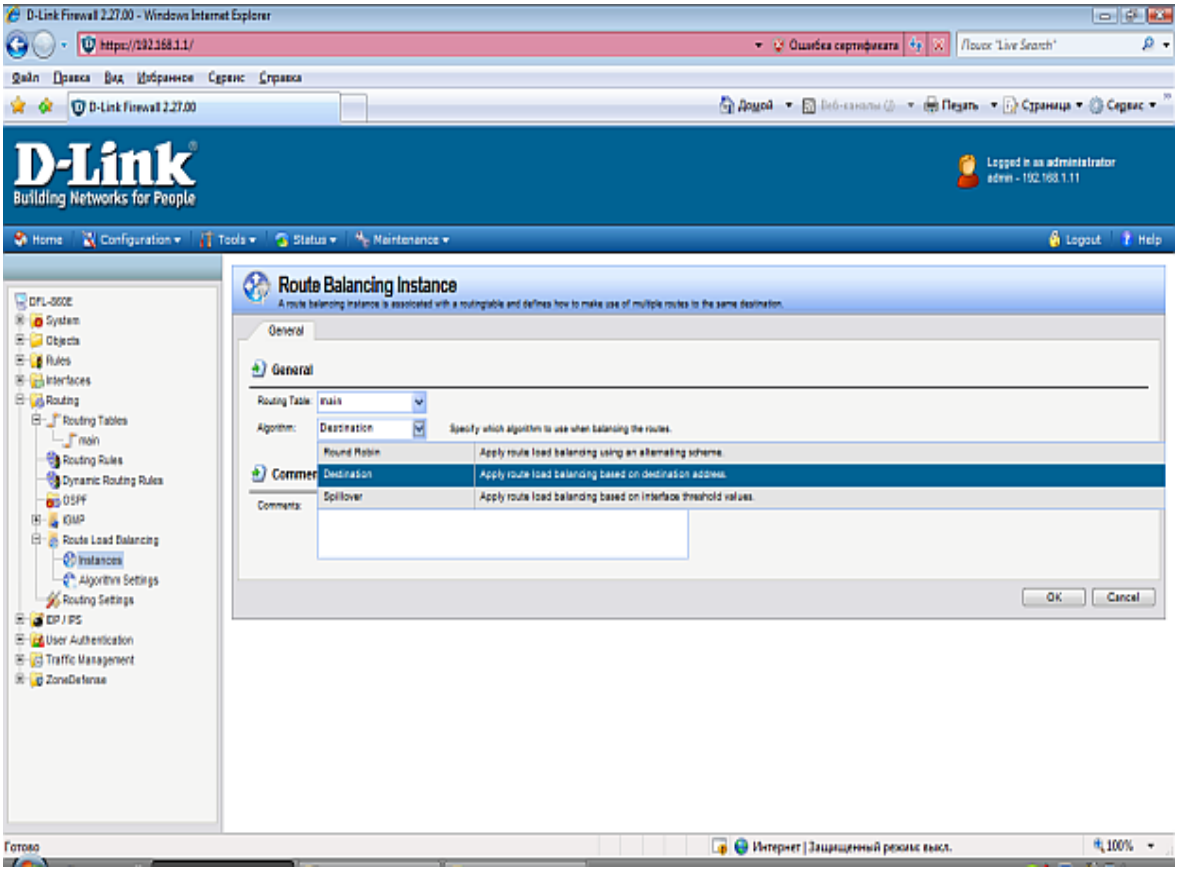

Рисунок 1.2 - Вибір алгоритму розподілу навантаження в міжмережевих екранах NetDefend

Є можливість вибрати один з алгоритмів розподілу навантаження між Інтернет-інтерфейсами:

- алгоритм RoundRobin розподіляє навантаження між інтерфейсами WAN1 і WAN2 послідовно (по черзі). Кожен раз, коли виникає нова виходить сесія з інтерфейсу LAN, вибирається інтерфейс WAN1 або WAN2 для відправки пакетів. Надалі, пакети даної сесії будуть використовувати раніше певний WAN-інтерфейс. TCP-сесія відкривається і закривається на одному і тому ж WAN-інтерфейсі;

- алгоритм Destination дозволить уникнути проблем з деякими протоколами при використанні балансування, наприклад FTP. Даний алгоритм працює аналогічно алгоритму Round Robin, за винятком того, що всі дані до віддаленого хосту йдуть через той інтерфейс, через який з'єднання було встановлено;

- значення Spillover визначає граничне значення навантаження для основного WAN-порту (Routing → Route Load Balancing > Algoritm Setings). При досягненні цього навантаження за вказаний період почне використовуватися другий WAN-порт (для нових сесій). Як тільки завантаження основного каналу впаде, нові сесії будуть відкриватися на ньому.

Для використання метрик маршруту з алгоритмом Round Robin метрика кожного маршруту за замовчуванням дорівнює нулю. При використанні взаємопов'язаних алгоритмів Round Robin і Destination можна встановлювати різні значення метрик, що дозволяють створити пріоритет вибору маршрутів. Маршрути з мінімальним значенням метрики будуть вибиратися частіше, ніж маршрути з більш високим значенням.

Якщо в сценарії з двома Інтернет-провайдерами (часто зустрічається вираз "ISP-провайдер", тобто Internet Service Provider) потрібно, щоб велика частина трафіку проходила через одне з ISP-підключень, то слід активувати RLB і призначити менше значення метрики для маршруту основного ISP-підключення (наприклад, 90) щодо другого (наприклад, 100).

Якщо завдання полягає у рівномірній балансуванню трафіку між двома Інтернет-провайдерами, то значення метрик для обох маршрутів слід призначати однакову.

Використання метрик маршруту з алгоритмом Spillover

При використанні алгоритму Spillover для кожного маршруту обов'язково повинна бути визначена метрика. В цьому випадку система NetDefendOS завжди вибирає маршрут з найнижчим значенням метрики. Алгоритм не призначений для роботи з однаковими метричними значеннями маршрутів, тому адміністратору слід встановлювати різні значення метрик для всіх маршрутів, до яких застосовується алгоритм Spillover.

Значення метрики визначає порядок, відповідно до якого трафік перенаправляється на інший маршрут після того, як для обраного маршруту перевищено допустимий значення переданого трафіку.

Можна створити кілька альтернативних маршрутів з різними метричними значеннями, для кожного з яких визначена порогова величина налаштувань алгоритму - Spillover Setting - для кожного інтерфейсу. Спочатку вибирається маршрут з мінімальною метрикою; після того як перевищено допустимий поріг налаштувань алгоритму, буде обраний наступний маршрут.

Якщо на всіх альтернативних маршрутах досягнуті порогові значення Spillover Setting, то маршрут не змінюється.

Значення метрики на інтерфейсах (маршрутах), які використовуються в балансуванні, має бути встановлено вище, ніж для інших інтерфейсів (маршрутів). Чим нижче значення метрики на інтерфейсі (маршруті), тим частіше цей інтерфейс (маршрут) буде використаний для встановлення з'єднання, щодо інтерфейсу (маршруту) з великим значенням метрики. Частка використання інтерфейсів (маршрутів) буде пропорційна різниці між значеннями метрик на цих інтерфейсах (маршрутах).

Балансування навантаження мережі і HA-кластеризація (резервування пристроїв) міжмережевих екранів NetDefend Високий рівень мережевої відмовостійкості досягається за рахунок використання двох міжмережевих екранів NetDefend: основного пристрою (master) і резервного пристрою (slave). Основний і резервний міжмережеві екрани взаємопов'язані і складають логічний HA-кластер.

Міжмережеві екрани NetDefend не підтримують балансування навантаження в HA-кластеризації пристроїв, тобто розподіл навантаження між ними не забезпечується, так як один пристрій завжди є активним (active), в той час як інша перебуває в режимі очікування (passive).

2 РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ АГРЕГУВАННЯ В ОПЕРАЦІЙНИХ СИСТЕМАХ НА БАЗІ LINUX

2.1 Приклад реалізації в OpenWrt

OpenWrt - вбудована операційна система, заснована на ядрі Linux, і призначена, в першу чергу, для домашніх маршрутизаторів. Основні компоненти включають в себе ядро Linux, util-linux, uClibc або musl і BusyBox. Розмір всіх компонентів оптимізований в зв'язку з тим, що в більшості домашніх маршрутизаторів сильно обмежений обсяг пам'яті.

Конфігурація OpenWrt проводиться за допомогою командного рядка (з оболонкою ash), набору скриптів UCI (Unified Configuration Interface - уніфікований інтерфейс розширеного налаштування), або заснованого на ньому веб-інтерфейсу LuCI (Lua Configuration Interface - інтерфейс конфігурації на Lua). У репозиторії є більше 3500 опціональних пакетів програм, доступні для установки за допомогою системи управління пакетами opkg.

Проект OpenWrt було розпочато в 2004 році після того, як Linksys створила прошивку для своїх бездротових маршрутизаторів популярної в той час серії WRT54G з відкритим вихідним кодом, ліцензованим по Стандартної громадської ліцензії GNU. Відповідно до умов цієї ліцензії Linksys повинна була зробити вихідний код своєї модифікованої версії доступним з тією самою ліцензією, що дозволило незалежним розробникам створювати свої похідні версії.

Спочатку підтримка обмежувалася серією LinksysWRT54G (LinksysWRT54Gseries), Але поступово розширилася і включає в себе чіпсети інших виробників, в тому числі і x86. Найбільш популярними в рамках проекту довгий час були серії LinksysWRT54G і AsusWL500G.

OpenWrt історично в основному використовує інтерфейс командного рядка, але однією з опцій є веб-інтерфейс, також надав згодом широкі можливості по конфігурації OpenWrt. Технічна підтримка традиційно здійснюється за допомогою форуму і IRC-каналу.

Головною відмінною рисою OpenWrt є повна підтримка файлової системи JFFS2, яка дозволяє використовувати для управління пакетами менеджер пакетів ipkg (в нових версіях opkg). Все це робить OpenWrt легко настроюється і адаптується системою для кожного конкретного випадку. У версіях для роутерів, що мають великий обсяг флеш-пам'яті (від 4 Мб), зазвичай використовується ФС SquashFS, яка використовує оверлей (суміщення змінюваних і незмінних файлів в одному каталозі). В такому випадку ФС менш ефективно використовує простір, так як зберігає в окремому розділі опису змін, але дозволяє легко зробити відкат до налаштувань за замовчуванням.

Стандартна прошивка надає базовий набір функцій. Для розширення функціональності використовуються додаткові пакети. Відзначається незручність веб-інтерфейсу (особливо для недосвідчених користувачів).

Завдяки можливості самостійної компіляції прошивки (в тому числі і ядра), використання OpenWrt дозволяє реалізувати практично всі відомі методи організації мереж. За замовчуванням в більшості готових офіційних «збірок» можливе використання таких методів:

- Static IP;
- DHCP Client;
- PPTP;
- PPPoE (втомучислі DualAccess PPPoE);
- UCI і LuCI.

Для традиційної настройки Unix-подібних систем необхідно заповнення великої кількості текстових файлів конфігурації, більшість яких має різний синтаксис, і виклик великої кількості утиліт командного рядка з різноманітними параметрами, що вимагає створення і налагодження досить складних скриптів (сценаріїв).

Замість цього OpenWRT пропонує уніфікований інтерфейс конфігурації UCI (UnifiedConfigurationInterface), що дозволяє управляти більшістю системних параметрів за допомогою єдиного синтаксису файлів конфігурації і командного рядка.

Файли конфігурації UCI знаходяться в гілці /etc/config і в загальному випадку не призначені для редагування користувачем. Для перегляду і зміни параметрів конфігурації служить утиліта uci. При виконанні команд uci set створюється тимчасова конфігурація, робоча ж конфігурація не змінюється. Команда uci revert скасовує зміни, зроблені в тимчасовій конфігурації, що не впливає на можливість робочої. Фактичне застосування конфігурації відбувається тільки по команді uci commit, яка переводить уніфіковане опис конфігурації у взаємно-пов'язане, несуперечливе стан традиційних файлів конфігурації і сценаріїв. При цьому автоматично перезапускає необхідні системні служби, що позбавляє від необхідності робити це вручну або перезавантажувати систему.

Стандартний web-інтерфейс LuCI використовує UCI для отримання відомостей про систему і внесення змін в її конфігурацію.

2.2 Реалізація агрегування в проекті X-Wrt та інші проекти, засновані на OpenWrt

Споріднений проект X-Wrt є розширенням OpenWrt для кінцевого користувача. OpenWrt є базовою системою з мінімальним веб-інтерфейсом для налаштування опцій. Основним розширенням X-Wrt є webif², веб-інтерфейс, який має близько 40 сторінок з опціями настройки роутера. Webif² включає в себе графіки мережевого трафіку і системного моніторингу, сторінки настройки та контролю мережі, бездротового з'єднання і безпеки. Налаштування передбачені для наступних сервісів: ведення логів, завантаження, cron, NVRAM, редагування тексту, управління ipkg, SNMP, резервне копіювання і відновлення, оновлення прошивки, WAN, VLAN, Wi-Fi, WEP, WPA, WDS, MACfiltering, Firewall, Portforwarding, DHCP, Dnsmasq, Hostnames, IPcontrol, Routing, UPnP, QoS, DynDNS, Wake-on-LAN, OpenVPN, PPTP і точка доступу Wi-Fi.

За 15 років існування проекту OpenWrt було реалізовано багато напрямів його розвитку, що дозволяє перекрити специфічні потреби у використанні пристроїв, що базуються на програмному забезпеченні Linux:

- PacketProtector - проект, заснований на OpenWrt і підтримує IDS, IPS, VPN;
- Gargoyle – проект, орієнтований на кінцевого користувача;
- Coova - OpenWrt-based проект, націлений на бездротові точки доступу;
- Freifunk - німецький проект, заснований на OpenWrt, доступний на кількох мовах;
- DebWrt - проект, націлений на запуск Debian на роутерах, підтримуваних OpenWrt;
- LEDE (Linux Embedded Development Environment) - проект, розпочатий частиною колишніх розробників OpenWrt, незгодних з політикою основної команди. Після 1,5 років розробки об'єднаний з OpenWrt.

2.3 Практичне використання LACP

На поточний час більшість рішень для агрегування гігабітних каналів ґрунтується на стандарті IEEE 802.3ad. Однак нестандартизовані протоколи інших фірм існували ще до прийняття цього стандарту, деякі з них використовуються досі. Ці протоколи в більшості своїй працюють виключно з продукцією однієї компанії або продукцією однієї лінії. Деякі з них мають певні переваги перед стандартом, наприклад EtherChannel, використовуваний Cisco, підтримує різні режими посилки пакетів, тоді як 802.3ad підтримує тільки стандартний режим. Серед інших нестандартних протоколів агрегації - Duralink Trunking (Adaptec), MLT (multi link trunking, Nortel).

До середини 2000-х років більшість виробників перейшли на випуск мережевих пристроїв з підтримкою стандарту IEEE 802.3ad, що в принципі має забезпечувати можливість спільної роботи пристроїв різних марок. Однак на

практиці деякі поєднання можуть виявитися непрацездатними, тому в специфікаціях часто спеціально уточнюється можливість спільної роботи тих чи інших пристроїв.

IEEE 802.3ad - це стандартний спосіб об'єднання декількох ліній зв'язку в одну. Принципово він нічим не відрізняється від технології EtherChannel, тобто кілька адаптерів Ethernet об'єднуються в один віртуальний адаптер, що забезпечує більш високу пропускну здатність і більш надійний захист від неполадок.

Наприклад, адаптери ent0 і ent1 можна об'єднати в об'єднання ліній IEEE 802.3ad ent3. Потім інтерфейсу ent3 можна призначити IP-адресу. З точки зору системи об'єднані адаптери є один адаптер. Протокол IP налаштовується для цих адаптерів як і для будь-якого іншого адаптера Ethernet [18].

Основне застосування технології агрегації - об'єднання каналів в мережевих комутаторах, але можна налаштувати агрегування для комп'ютерних мережевих адаптерів. Наприклад, в операційній системі Linux можна конфігурувати агрегований мережевий адаптер bond0 зі стандартним драйвером ядра (англ. Bonding driver) як об'єднуючий Ethernet-адаптери eth0 і eth1, з призначенням йому єдиного IP-адреси, і для системи і виконуваних на ній програм немає ніякої різниці між таким адаптером та фізичними (виключаючи деякі службові утиліти, які призначені для операцій безпосередньо з адаптерами). При цьому значення MAC-адреси bond0 будуть чергуватися - періодично буде показуватися то MAC-адресу першої мережевої карти eth0, то MAC-адресу адаптера eth1.

У загальному випадку, агрегування восьми стандартних каналів за допомогою 802.3ad виявляється дешевше, ніж один пристрій, що підтримує на порядок більшу пропускну здатність, і дозволяє поступово збільшувати швидкість каналів в системі без необхідності купувати дорогі швидші адаптери. Однак, агрегування має обмеження: розподіл трафіку по каналах може бути нерівномірним, аж до того, що весь трафік йде по одному каналу, а інші простоюють (залежить від трафіку, можливостей і налаштувань обладнання), що

в крайніх випадках означає відсутність виграшу в пропускної здатності в порівнянні з єдиним каналом. Крім того, об'єднувати можна не більше восьми каналів, що в разі гігабітних каналів дає теоретичну сумарну пропускну здатність лише в 8 Гбіт / сек замість 10 Гбіт / сек, які може забезпечити один швидкодіючий адаптер.

Як правило, всі порти при агрегування повинні бути одного типу, наприклад, всі порти для кручений пари, все - для одномодового оптоволокна (SM) або все - для багатомодового оптоволокна (MM). Об'єднуються порти повинні бути налаштовані на одну швидкість передачі (хоча за стандартом 802.3ad змішувати порти з різною швидкістю допустимо, на практиці такі конфігурації часто виявляються непрацездатними).

2.4 Балансування трафіку LAG

Балансування трафіку здійснюється за допомогою вибору фізичного каналу відправником фрейма за допомогою обраного алгоритму. До основних і часто використовуваних можна віднести наступні алгоритми:

- по MAC-адресу відправника або MAC-адресу одержувача або з огляду на обидві адреси;
- по IP-адресою відправника або IP-адресою одержувача або з огляду на обидві адреси;
- за номером порту відправника або номеру порту одержувача або з огляду на обидва порту.

Як приклад наведемо два агрегованих з'єднань при використанні методу балансування по мак-адресою відправника. В даному випадку індексом для балансування буде використовуватися останній біт мак-адреси відправника, що представлено на рисунку 2.1.

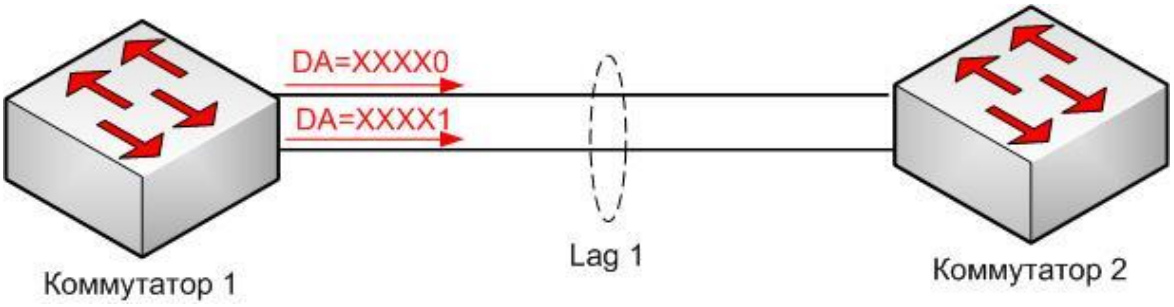

Рисунок 2.1 - З’єднання з 2 каналів

У разі, якщо лінків буде 4, то для балансування буде використовуватися останні 2 біти мак-адреси, як показано на рисунку 2.2.

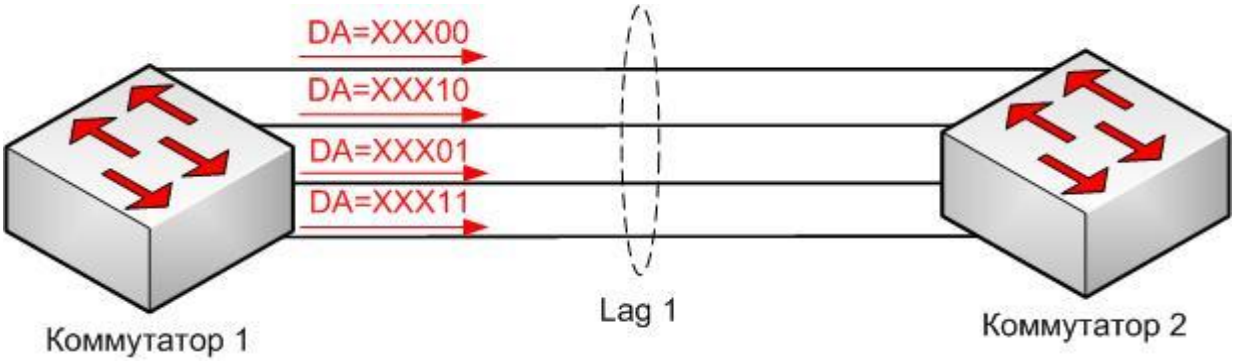

Рисунок 2.2 - З’єднання з 4 каналів

Відповідно, якщо в лагу буде 3 лінка, то як можна здогадатися при використанні даного методу рівномірного балансування домогтися буде складно і піде перекіс по трафіку на будь-якої лінк. Тому слід ставитися уважно до вибору методу балансування.

2.5 Реалізація підсистеми LINUX NETWORKING

В даний час стек Linux більше не є самостійною операційною системою і обслуговує різні функції по мережі. Кількість та тип додатків, які підтримує мережний стек, може варіюватися від андроїд них апаратів до маршрутизаторів і комутаторів центрів обробки даних, як віртуалізованих, так і в апаратній

реалізації. Додаток є ознакою сучасного різноманіття. Деякі з них орієнтовані на кінцевого споживача, інші – промислово-орієнтованих. Існує багато різних спектрів додатків, і коли ви маєте різноманітність у вашому прикладному просторі, важко мати одне мережеве рішення. Це чинило тиск на мережу Linux, щоб розвиватися і підтримувати різноманітні стеки програм з різними мережевими вимогами. Виклик виникає внаслідок різних очікувань кінцевих вузлів від очікування середнього вузла під управлінням Linux. Стек Linux повинен працювати по-різному у всіх цих областях.

2.5.1 LinuxNetworking і центри обробки даних

Linux є продуктивним в центрах обробки даних і є базою для відкритих мережевих середовищ. Багато функцій віртуального комутатора доступні з апаратним вивантаженням для прискореної роботи [19]. Ядро Linux підтримує три типи програмних мостів - Bridge, MACVLAN і OpenvSwitch. Існує також вбудований комутатор NIC з SR-IOV, який може використовуватися замість програмного комутатора. Останнім часом було багато нових функцій моста, таких як FDB маніпуляція, VLAN-фільтрація, Bridge, фільтрація VLAN для 802.1ad (Q-in-Q).

Типовий конвеєр обробки пакетів комутатора включає:

- Розбір і класифікація пакетів - L2, L3, L4, тунелювання, VXLANVNI, внутрішній пакет L2, L3, L4;
- Push / pop для VLAN або інкапсуляції / декапсуляції для тунелювання;
- Функція, пов'язана з QoS, наприклад, вимірювання, формування, маркування та планування;
- Операції перемикання.

Обробка даних прискорюється шляхом розкладання конвеєра обробки пакетів і вивантаження деяких етапів на апаратні ASIC. Можливості рівня 2, які можуть бути перевантажені в ASIC, можуть включати в себе накопичення MAC і

їх зберігання, обробку STP, відслідковування IGMP і VLXAN. Також можливо розвантажити функції рівня 3 до ASIC.

2.5.2 Linux Switch Types

Bridge - це стандартний міст MAC і VLAN, що містить FDB (Forwarding Data Base), STP (spanning tree) і IGMP (Internet Group Management Protocol) функції. Bridge містить запис MAC для розподілу портів у FDB. Побудова FDB називається «навчання МАК» або просто «процес навчання».

MACVLAN - це комутатор на основі STATIC MAC & VLAN. Він використовує одноадресну фільтрацію замість безладного режиму і підтримує ряд режимів - приватний, VEPA, bridge і passthru. MACVLAN є зворотним VLAN під Linux. Він приймає один інтерфейс і створює кілька віртуальних інтерфейсів з різними MAC-адресами. По суті, це дає можливість створювати незалежні логічні пристрої по одному пристрою Ethernet - відносини «багато до одного» на відміну від відносин «один на багато», коли ви перетворюєте одну мережу на декілька мереж. MACVLAN пропонує ізоляцію в тому сенсі, що він буде бачити трафік тільки на інтерфейсі з вказаною MAC-адресою.

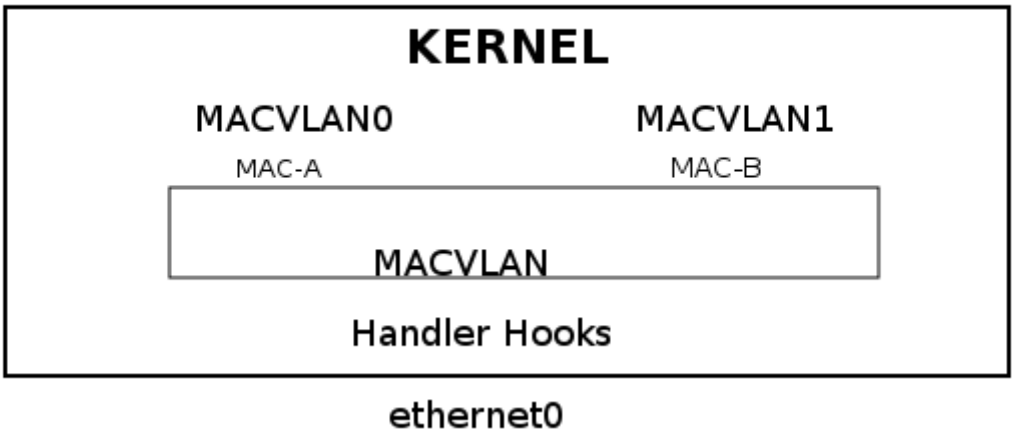

Рисунок 2.3 - Структура комутатора MACVLAN

OpenvSwitch - це NOS на основі мостів linux. Він підтримує протоколи STP і, що важливіше, OpenFlow. Її комутація базується на потоках і все пересилається на основі таблиці потоків. Він також використовується у багатьох складних випадках використання вкладених проектів OpenvSwitch з OVN (віртуальні мережі OpenSource). За замовчуванням, OVS діє, як звичайний комутатор Layer 2. Для розширених операцій він може бути підключений до контролера SDN або використовувати командний рядок для додавання правил OpenFlow вручну.

2.5.3 Мережі Linux і Android

Linux широко використовується як база для пристроїв на Android. Мережевий стек Linux має різні потреби для мобільних пристроїв, ніж для пристроїв центрів обробки даних. Мобільний пристрій весь час рухається, підключається до різних мереж різної якості, тобто підключені до декількох мереж майже весь час. Якщо пристрої знаходяться в мережі WIFI і, наприклад, потребують відправки SMS, потрібно відкрити мережу стільникового зв'язку, яка знаходиться на іншому інтерфейсі IP. Користувачі хочуть одночасно всі мережі, а стек Linux повинен безперешкодно перемикатися через межі мережі. Для цього програма повинна контролювати всі TCP-з'єднання, щоб вони не були заблоковані вроботі, та забезпечувати неможливість їхньої конкуренції між собою. Як правило, в Linux при видаленні IP-адреси з'єднання TCP залишатиметься там, сподіваючись, що IP-адреса повернеться. В результаті для кожного мережевого комутатора з'єднання TCP закриваються.

Linux також повинен підтримувати різні маршрутизацію для кожного додатка і сокета, наприклад, підключення до бездротового принтера під час роботи в мережі CELL. Існує також спосіб, за допомогою якого користувачі можуть дізнатися, чи підключаються вони до мережі WIFI, яка не має з'єднання між кодами. Для цього Linux необхідно використовувати DNS і відкрити TCP-з'єднання на мережі backhaul. Для таких, як невеликі пристрої, мережевий стек повинен обробляти багато різних функцій.

2.5.4 Реалізація на Linux Networking Subsystem

Архітектура системи Linux містить простір користувача, ядро і фактичне обладнання. У верхній частині рамки Linux існує простір користувача з різними користувачами. У середині ядра простору пересилають пакети, приймаючи інструкцію з елемента простору користувача. У самому низу ми маємо власне обладнання, таке як процесор, оперативна пам'ять і мережева плата. Один із способів спілкування між користувачами та ядром здійснюється через Netlink. Роз'єм Netlink - це те, що обробляє двонаправлену комунікацію між ними. Вона може бути створена в просторі користувача за допомогою системного виклику socket () і або в ядрі за допомогою netlink_kernel_create (). Нижче наведено сокет Netlink, створений у ядрі та просторі користувачів.

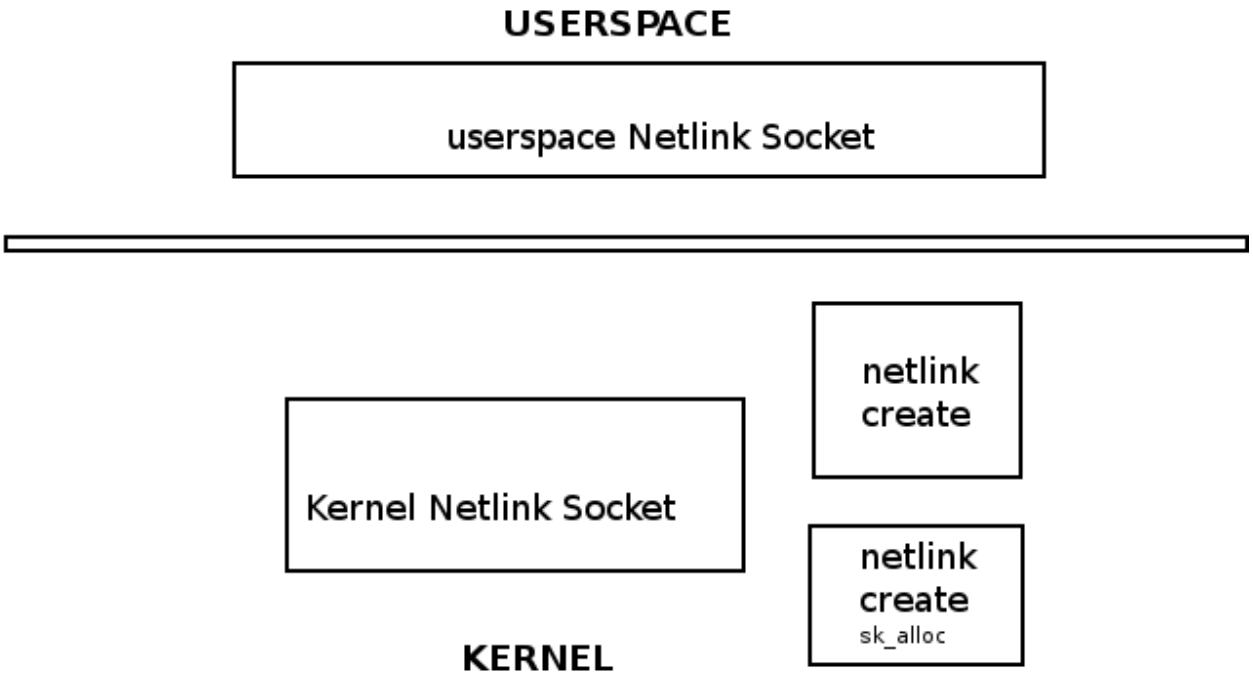

Рисунок 2.4 - Схема ядра Linux Networking Subsystem

Реалізація протоколу Netlink знаходиться в наступній папці net / netlink, наведеної нижче. Af_netlink надає мережевий інтерфейс Socket API, genetlink надає загальний netlink API і diag надає інформацію про сокетах netlink.

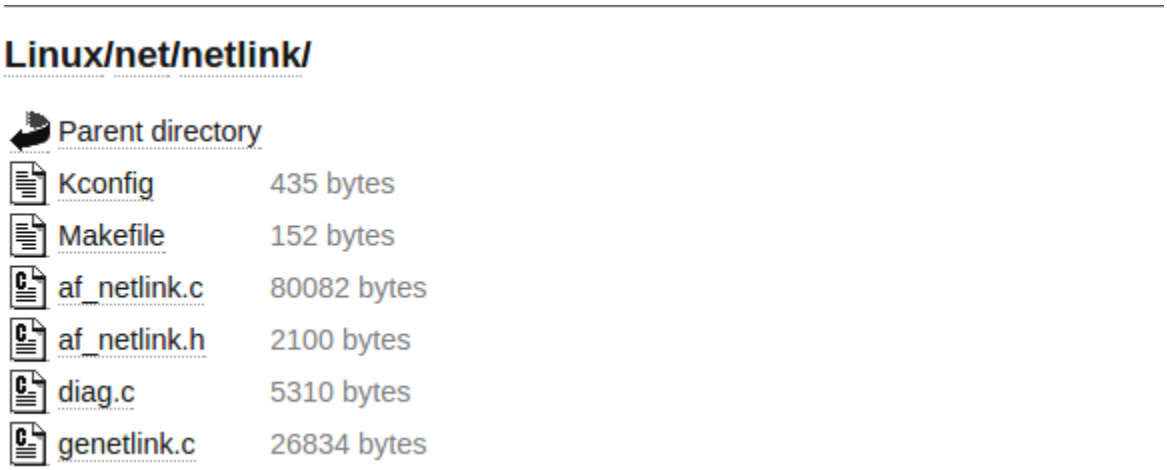

Рисунок 2.5 - Файли реалізації протоколу Netlink

Мережеві підсистеми Linux є частиною простору ядра і є однією з найважливіших підсистем. Навіть якщо хости не підключені, мережева підсистема використовується для взаємодії клієнт-сервер X-Windows.

Стек мережевого стеку Linux обробляє вхідні пакети, що надходять на рівень 2, до мережного рівня, а потім передає для локальної доставки до протоколів транспортного рівня, слухаючи сокети TCP або UDP. Будь-які пакети, не призначені для локальної системи, відправляються назад по стеку для передачі. Ядро не обробляє нічого над рівнем 4. Всі шари над рівнем 4 обробляються додатками Userspace.

2.5.5 Структури sk_buff і net_device

Sk_buff і net_device є фундаментальними для мережевої підсистеми. Драйвер мережевого пристрою (структура net_device) приймає і передає пакети, або спрямовує їх в стек (Layer 3 toLayer 4), або передавае на вихідний інтерфейс. Для визначення інтерфейсу і конкретної діяльності з обробки пакетів

здійснюється пошук в підсистемі маршрутизації для кожного вхідного / вихідного пакета. Існує багато речей, які можуть вплинути на обхід пакетів, таких як регулятори Netfilter, IPsec підсистема, TTL і т.д. Пакети приймаються по лінії NIC (netdevice) і поміщаються в sk_buff, а потім передаються через мережевий стек.

Стек користувачів мережевого простору може сповільнити роботу ЦП. Все, що переходить до ядра, впливає на продуктивність. Отже, якщо програма перетне межу користувача / ядра, то вона буде перегружати систему. Потрібно це мінімізувати, зберігаючи якомога більше в ядрі і нижче, і переходити до простору користувача для більшої розгрузки. Наприклад, транзитний трафік може не вимагати постійного користування.

3 АЛГОРИТМ ТА РЕАЛІЗАЦІЯ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ КАНАЛІВ ЗВ’ЯЗКУ VOL

3.1 Опис архітектури системи

Система балансування трафіку складається з трьх підсистем з однозв’язними залежностями:

- підсистема розподілу трафіку. Відповідає за рівномірне розподілення кадрів між каналами, що перебувають у черзі на відправку. Код виконується у просторі ядра та має максимально спрощену архітектуру для того, щоб не сповільнювати роботу. Не залежить від підсистем в просторі користувача та може працювати самостійно;

- підсистема визначення стану каналів. Працює у просторі користувача. Відповідає за визначення числової характеристики кожного каналу, спираючись на задані критерії. В якості критеріїв може бути використаний час проходу пакета від клієнта до сервера та від сервера до клієнта. Не залежить від модуля ядра. Може виконувати моніторинг каналів без балансування;

- підсистема визначення пріоритетів. Це код, що виконується у просторі користувача та взаємодіє з підсистемою визначення стану каналів та підсистемою розподілу трафіка. Підсистема збирає інформацію про стан каналів та передає у модуль ядра (підсистему розподілу трафіку).

Зв’язки між підсистемами зображені на рисунку 3.1.

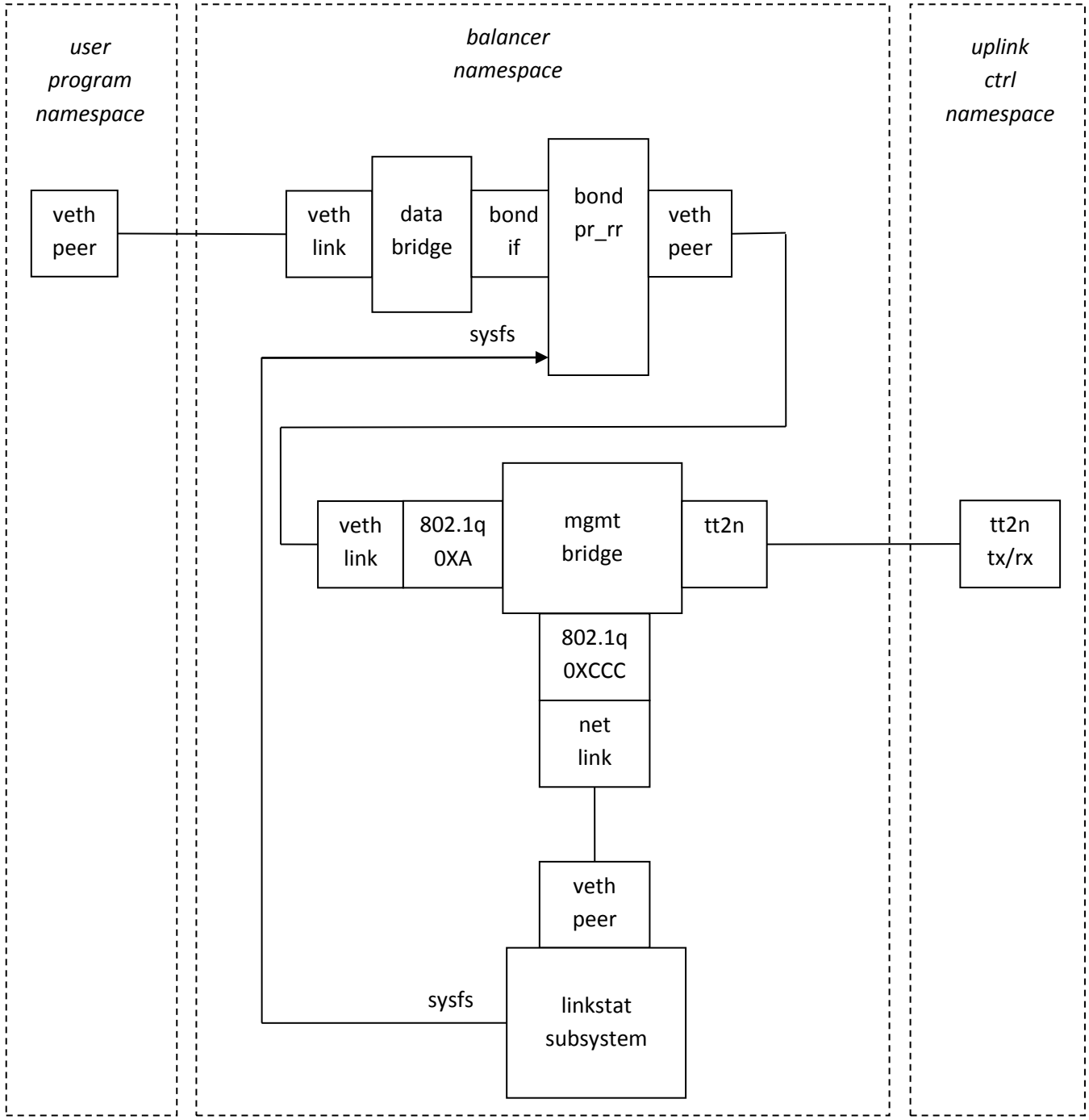

Рисунок 3.1 – Структурна схема взаємодії програмних компонентів

3.2 Підсистема розподілу трафіку

Підсистема балансування трафіку на основі пріоритетів реалізована в якості завантажуваного модуля ядра Linux. Даний механізм дозволяє додавати функціонал у ядро під час його виконання та зберігає допустимий розмір монолітного ядра під час запуску.

Одією з ідеологій цього проекту було максимальне повторне використання існуючого коду, функціоналу та можливостей. Данний принцип повинен бути закладений у всі додатки для юнікс-подібних систем.

В дереві першокоду Linux був обраний модуль, що реалізує функції балансування трафіку та має багаті можливості по вибору алгоритмі розподілу трафіку в черзі на відправку (по хеш-функції MAC-адреси IP, випадково, та ін.).

Для поставленного завдання набільш ближче був алгоритм round-robin (англ. карусель). Цей алгоритм може працювати у декількох режимах:

- випадкови розподіл пакетів з черги на відправку;
- рівномірний розподіл пакетів з черги на відправку.

Останній режим може працювати з будь якою кількістю пакетів на канал. Тобто задавши параметр packet_per_slave у sysfs наприклад 5 – в канал буде відправлено 5 пакетів до того, як алгоритм почне відрправляти пакети в інший канал.

Приклад налаштування параметру packet_per_slave наведено нижче.

```
root@build:/sys/class/net/bond0/bonding# ip l
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default qlen 1000
    link/ether 3a:0c:8e:95:ee:6a brd ff:ff:ff:ff:ff:ff
```

```
    3: veth0@veth1: <NO-CARRIER,BROADCAST,MULTICAST,SLAVE,UP,M-DOWN> mtu
1500 qdisc noqueue master bond0 state LOWERLAYERDOWN mode DEFAULT group
default qlen 1000
        link/ether 3a:0c:8e:95:ee:6a brd ff:ff:ff:ff:ff:ff
    4: veth1@veth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
mode DEFAULT group default qlen 1000
        link/ether 8e:c4:f2:ae:2e:b6 brd ff:ff:ff:ff:ff:ff
    5: veth2@veth3: <NO-CARRIER,BROADCAST,MULTICAST,SLAVE,UP,M-DOWN> mtu
1500 qdisc noqueue master bond0 state LOWERLAYERDOWN mode DEFAULT group
default qlen 1000
        link/ether 3a:0c:8e:95:ee:6a brd ff:ff:ff:ff:ff:ff
    6: veth3@veth2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
mode DEFAULT group default qlen 1000
        link/ether ca:d8:9d:46:b8:07 brd ff:ff:ff:ff:ff:ff
```

В загальному вигляді цей алгоритм працює наступним чином: відправляється задана кількість пакетів в перший канал, потім така ж кількість пакетів відправляється у другий і т.д.

Нажаль пропускна можливість ніде не враховується, тому алгоритм може ефективно працювати тільки з каналами з однаковими характеристиками.

Оскільки вибір каналу відбуває на основі лічильнику відправлених пакетів, а не вмісту пакету – це гарантує рівномірний розподіл, незалежно від виду трафіку.

Через свою схожість реалізованого алгоритму з реалізацією rounf-robin він був названий pr_rr (priorities round-robin).

Алгоритм pr_rr працює наступним чином:

- у кожного каналу в LAG є свій відносний пріорітет (ge1 – 2, ge2 – 4 та ge1 – 8, ge2 – 16 це аналогічні конфігурації: в обох випадках в ge2 буде навантаження в два рази вище, ніж ge1);

Приклад налаштування параметру pr_rr_prio наведено нижче.

```
root@build:/sys/class/net/bond0# ls
addr_assign_type   carrier                dev_port              ifalias
lower_veth2       phys_port_id    queues        type
addr_len            carrier_changes      dormant               ifindex
mtu               phys_port_name  speed         uevent
address             carrier_down_count   duplex                iflink
name_assign_type  phys_switch_id  statistics
bonding             carrier_up_count     flags                 link_mode
netdev_group      power           subsystem
broadcast          dev_id               gro_flush_timeout  lower_veth0
operstate         proto_down      tx_queue_len

root@build:/sys/class/net/bond0/bonding# ls
active_slave         ad_num_ports        all_slaves_active   downdelay
miimon        packets_per_slave  resend_igmp     xmit_hash_policy
ad_actor_key       ad_partner_key    arp_all_targets    fail_over_mac
min_links     pr_rr_prio         slaves
ad_actor_sys_prio   ad_partner_mac     arp_interval          lacp_rate
mode          primary            tlb_dynamic_lb
ad_actor_system    ad_select          arp_ip_target       lp_interval
num_grat_arp  primary_reselect   updelay
ad_aggregator        ad_user_port_key   arp_validate         mii_status
num_unsol_na  queue_id           use_carrier
root@build:/sys/class/net/bond0/bonding# cat mode
balance-pr-rr 7


root@build:/sys/class/net/bond0/bonding# cat pr_rr_prio
veth0 8
veth2 4
root@build:/sys/class/net/bond0/bonding# echo veth0 16 >pr_rr_prio
root@build:/sys/class/net/bond0/bonding# cat pr_rr_prio
veth0 16
veth2 4
root@build:/sys/class/net/bond0/bonding# echo veth2 8 >pr_rr_prio
root@build:/sys/class/net/bond0/bonding# cat pr_rr_prio
veth0 16
veth2 8
```

- На початку роботи алгоритму усі числові значення пріоритетів на всіх каналах сумуються;

- Обирається випадкове число від нуля до того, що було обчислено на попередньому кроці;

- Алгоритм проходить по всім каналам у списку, акумулює їх значення пріорітетів та порівнює з випадковим числом з попереднього кроку. Як тільки акумульована сума перевищила випадкове число – пакет відправляється у канал.

Випадковий характер розподілу пакетів по каналам був обраний так, як усі канали є нестабільними по пропускній можливості. Їхні характеристики можуть змінюватись у випадковому порядку. Робота саме з такими каналами була ціллю данного проекту. Випадковий характер розподілу дає можливість анігілювати випадковий характер змін пропускних можливостей каналів, на які не встигла відреагувати підсистема визначення стану каналів та підсистема визначення пріоритетів.

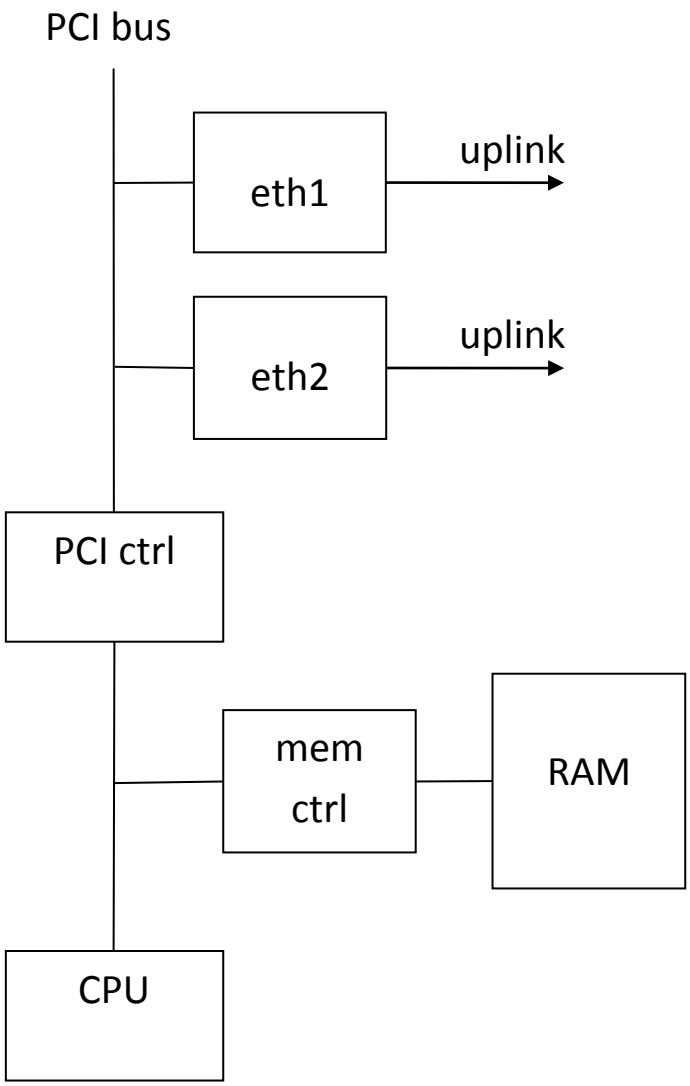

Рисунок 3.2 – Схема використання технології DMA для пересилання пакетів

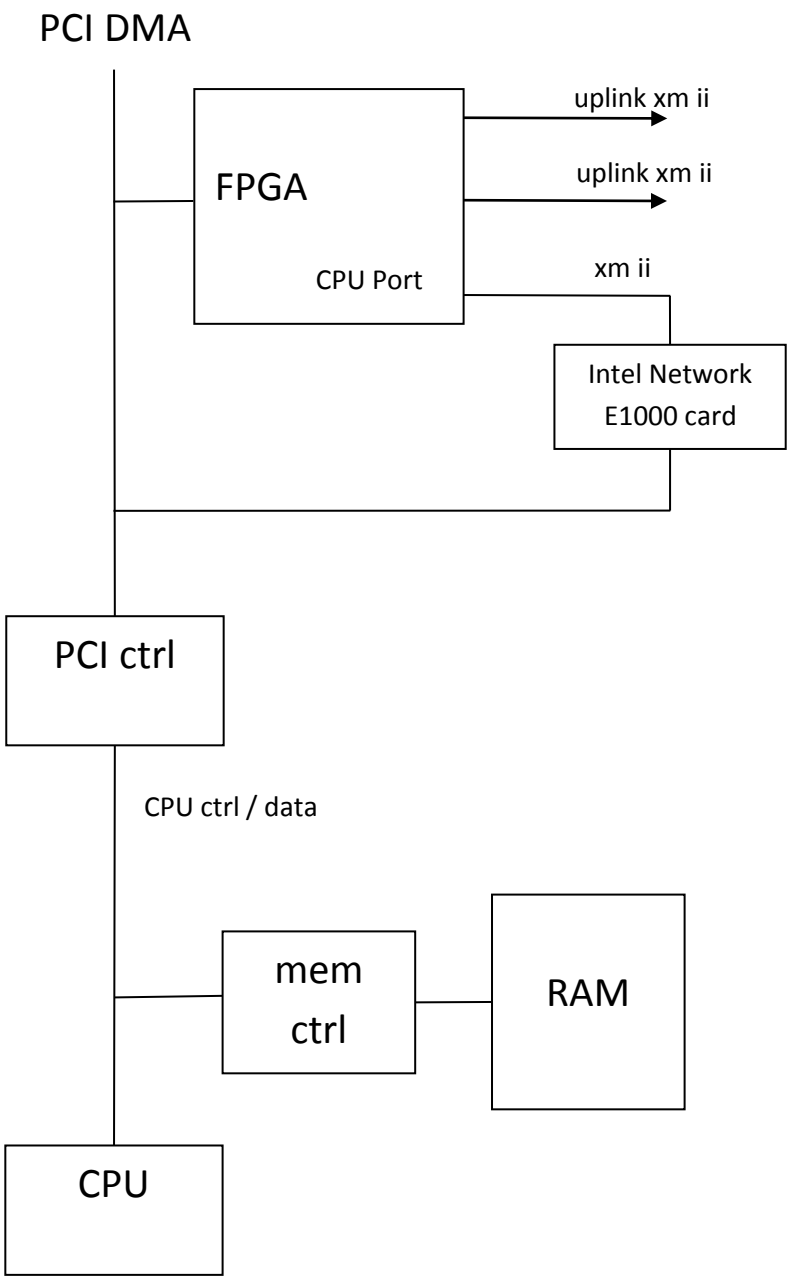

Рисунок 3.3 – Схема використання технології FPGA для пересилання пакетів

3.3 Підсистема визначення стану каналів

Підсистема визначення стану каналів реалізована у вигляді программи, якак виконується у просторі користувача. Таке рішення було прийнято для зменшення кодує що виконується у ядрі.

Швидкодія даного елемента не є критичним параметром. Для визначення пропускної спроможності нестабільного каналу необхідно зробити декілька вимірів параметра, на якому базується пропускна спроможність (наприклад час проходження пакету від клієнта до серверу та навпаки). Для розробки концептуальної моделі даної системи було використано метод визначення характеристик каналу на базі часу проходження пакету по ньому. Виміри ініціює клієнтська частина системи. Серверна частина виставляє пріоритети на відповідних каналах згідно інформації, яка була отримана від клієнта. Такий режим роботи був прийнятий так, як збої в передачі даних виникають ближче до клієнта, тому і визначення помилок та збоїв в передачі логічно реалізувати на стороні клієнта.

Для передачі службової інформації використовується то же канал, що і для даних. Виникає необхідність розділяти трафік з даними та сервісний трафік до входу в систему балансування та відправляти після неї (для того, щоб сервісні підсистеми могли контролювати, по якому каналу буде відправлений пакет).

Розділення сервісного трафіку та трафіку користувача відбувається завдяки використанню технології 802.1q. На кожний пакет користувача навішується мітка з тегом 0xA, а на сервісний пакет – 0xCCC. Перед відправкою на підсистему балансування трафік проходить через міст, на якому трафік з міткою 0xCCC перенаправляться на менеджмент інтерфейс (без мітки), а трафік з тегом 0xA перенаправляється на інтерфейс, що підключений до підсистеми балансування.

Пакет з даними перед відправкою з середовища підсистеми балансування:

09:41:52.372384 00:56:66:98:34:99 > 00:13:95:2f:fd:a4, ethertype 802.1Q (0x 8100), length 90: vlan 10, p 0, ethertype IPv4, (tos 0x0, ttl 64, id 1, off set 0, flags [none], proto Options (0), length 72)

192.168.2.1 > 192.178.1.2: ip-proto-0 52

0x0000: 0013 952f fda4 0056 6698 3499 8100 000a

0x0010: 0800 4500 0048 0001 0000 4000 f657 c0a8

0x0020: 0201 c0b2 0102 3132 3334 3537 396f 6f6f

0x0030: 646a 6468 6469 6462 646a 6d64 6269 6562

0x0040: 6475 646e 6468 6a64 6263 6a64 6e64 6872

0x0050: 6a72 6263 6a64 6264 6e30

Пакет з сервісною інформацією перед відправкою з середовища підсистеми балансування:

09:42:54.980339 00:56:66:98:34:99 > 00:13:95:2f:fd:a4, ethertype 802.1Q (0x 8100), length 80: vlan 3276, p 0, ethertype IPv4, (tos 0x0, ttl 64, id 1, o ffset 0, flags [none], proto Options (0), length 62)

192.168.2.1 > 192.178.1.2: ip-proto-0 42

0x0000: 0013 952f fda4 0056 6698 3499 8100 0ccc

0x0010: 0800 4500 003e 0001 0000 4000 f661 c0a8

0x0020: 0201 c0b2 0102 5955 4255 4759 4759 4259

0x0030: 5659 5459 4656 5659 5654 4354 4354 4654

Приклад налаштування середовища для роботи підсистеми визначення стану каналів.

```
#!/bin/bash
set -o xtrace

# Create bridge with one trunk port and two access (veth)
# (peer)---(untagged bridge side)---(tagged_bridge_side)---(bridge)
function create_bridge_link_ctrl() {
 uplink_name=$1
 mgmt_vlan=$2
 data_vlan=$3
 br_name=mbr"$uplink_name"

 mgmt_link_peer=$4                  #peer-side link name
 mgmt_link_bt="$mgmt_link_peer"0bt #tagged bridge-side link name
 mgmt_link_bu="$mgmt_link_peer"0bu #untagged bridge-side link name

 data_link_peer=$5                  #peer-side link name
 data_link_bt="$data_link_peer"0bt #tagged bridge-side link name
 data_link_bu="$data_link_peer"0bu #untagged bridge-side link name

 ip link add $mgmt_link_bu type veth peer name $mgmt_link_peer
 ip link add link $mgmt_link_bu $mgmt_link_bt type vlan proto 802.1Q id
$mgmt_vlan

 ip link add $data_link_bu type veth peer name $data_link_peer
 ip link add link $data_link_bu $data_link_bt type vlan proto 802.1Q id
$data_vlan

 brctl addbr $br_name
 brctl addif $br_name $mgmt_link_bt
 brctl addif $br_name $data_link_bt
 brctl addif $br_name $uplink_name

 ip link set $br_name up
 ip link set $uplink_name up
 ip link set $mgmt_link_bt up
 ip link set $mgmt_link_bu up
 ip link set $mgmt_link_peer up
 ip link set $data_link_bt up
 ip link set $data_link_bu up
 ip link set $data_link_peer up
}

#Create lag bridge (pr_rr_mode)
# (databr_peer_link_name)--{|namespace border|}--(databr_link_name)-
(data_bridge)-(bond)
function create_pr_rr_lag_bridge() {
 databr_peer_link_name=$1
 bond_name=$2
 databr_name=db"$bond_name"
 databr_link_name="$databr_name"0dl
```

```
#create bond
 ip link add name $bond_name type bond
 echo balance-pr-rr >/sys/class/net/$bond_name/bonding/mode
 ip link set $bond_name up

 #create databridge
 ip link add $databr_link_name type veth peer name $databr_peer_link_name
 brctl addbr $databr_name
 brctl addif $databr_name $bond_name
 brctl addif $databr_name $databr_link_name
 ip link set $databr_name up
 ip link set $databr_link_name up
 ip link set $databr_peer_link_name up
}

#Attach uplink to created lag bridge (including mgmt/data ctrl bridge)
function attach_uplink_to_lag() {
 bond_name=$1
 uplink_name=$2
 mgmt_link_name=$3

 mgmt_vlan=3276 # 0xCCC
 data_vlan=10   # 0xA
 data_link_name="$uplink_name"0ol

 create_bridge_link_ctrl $uplink_name $mgmt_vlan $data_vlan
$mgmt_link_name $data_link_name
 ip link set dev $data_link_name master $bond_name
}


############################## MAIN ##############################
datalink_name=$1
bond_name=bn"$datalink_name"

#create bond with databridge
create_pr_rr_lag_bridge $datalink_name $bond_name

#Attach uplinks to bond
#Read uplinks
links_file=`mktemp`
echo '' >$links_file
while read link; do
 mgmt_link_name="$link"0mm
 attach_uplink_to_lag $bond_name $link $mgmt_link_name
 echo $link $mgmt_link_name >>$links_file
done

#This should be passed to balancer subsystem
echo $links_file
```

ВИСНОВОК

У ході виконання дипломного проекту була розроблена система, що дозволяє балансувати трафік через канали зі змінними характеристиками.

Архітектура системи модульна - будь який компонент (підсистема балансування, підсистема визначення стану каналів, підсистема визначенря пріорітетів) може бути покращений незалежно від інших.

Сучасні L2/L3 мости розроблені на базі так званого PacketCPU, що відіграє роль виконавчого пристрою. Центральний процессор виконує виключно сервісні функції (відправка та прийом BPDU, моніторинг стану каналів та ін.).

Оскільки данні пристрої розробляються як закриті системи - специфікації для керування з центрального процесора теж тримаються в таємниці.

Підсистема балансування розроблена через необхідність виконавчого пристрою, але неможливість отримати такий через закритість технології.

Альтернативну роль виконавчого пристрою може виконувати сучасна архітектура обчислювальних птистроїв загального призначення. Для того, щоб увесь трафік не проходив через центральний процессор, тим самим равантажуючи його - використовуються технологія DMA (Direct Memory Access). Цей метод передачі данних від мережевої карти прямо в память широко використовується на сьгоднішній день для програмної реалізації мостів. Один з таких - модуль bridge ядра Лінукс. Аналогічно влаштований модуль bonding.

Саме ці підсистеми ядра було вдосконалені для програмної реалізації виконавчого пристрою в даному проекті.

Розробка виконана з перспективою на перенесення коду для інших архітектур.

Тобто при необхідності підсистема балансування може бути перенесена на системи з ASIC або FPGA, що було продемонстровано в третьому розділі данної роботи.

ПЕРЕЛІК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Журнал «Коммунист», 1962г. , № 12. Статья «Информация и техника».
2. Олифер В. Г., Олифер Н. А. Глава 13. Коммутируемые сети Ethernet // Компьютерные сети. Принципы, технологии, протоколы. — 4-е изд. — СПб.: «Питер», 2010.
3. Understanding EtherChannel Load Balancing and Redundancy on Catalyst Switches [Електронний ресурс] : Режим доступу: https://www.cisco.com/c/en/us/support/docs/lan-switching/etherchannel/12023-4.html - Назва з екрану. 12.05.2019р.
4. IEEE P802.3ad Link Aggregation Task Force [Електронний ресурс] : Режим доступу: http://www.ieee802.org/3/ad/ - Назва з екрану. 18.05.2019р.
5. Link Aggregation Over View of the IEEE 802.3ad-2000 (clause 43) [Електронний ресурс] : Режим доступу: https://slideplayer.com/slide/3130142/ - Назва з екрану. 25.05.2019р.
6. Link Aggregation and LACP basics [Електронний ресурс] : Режим доступу: https://www.thomas-krenn.com/en/wiki/Link_Aggregation_and_LACP_basics - Назва з екрану. 25.05.2019р.
7. Catalyst 3750-X and 3560-X Switch Software Configuration Guide, Release 12.2(55)SE [Електронний ресурс] : Режим доступу: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/12-2_55_se/configuration/guide/3750xscg/swethchl.html - Назва з екрану. 22.05.2019р.
8. Configuring IEEE 802.3ad Link Bundling [Електронний ресурс] : Режим доступу: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cether/configuration/xe-3s/ce-xe-3s-book/ce-ieee-link-bndl-xe.pdf - Назва з екрану. 10.05.2019р.
9. Технология NIC Teaming в Windows Server 2012 [Електронний ресурс] : Режим доступу: https://windowsnotes.ru/windows-server-2012/texnologiya-nic-teaming-v-windows-server-2012/ - Назва з екрану. 02.06.2019р.

10. Введение в агрегирование каналов (NIC Teaming) [Електронний ресурс] : Режим доступу: http://www.k-max.name/vmware/razbiraemsya-s-lacp-i-nic-teaming-v-vmware/ - Назва з екрану. 03.06.2019р.

11. LAG (Link Aggregation Group) & LACP (Link Aggregation Control Protocol) – An Intro [Електронний ресурс] : Режим доступу: https://www.excitingip.com/3015/lag-link-aggregation-group-lacp-link-aggregation-control-protocol-an-intro/ - Назва з екрану. 03.06.2019р.

12. Configuring Link Bundling [Електронний ресурс] : Режим доступу: https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k_r5-2/interfaces/configuration/guide/b-interfaces-cg52x-asr9k/b-interfaces-cg52x-asr9k_chapter_01000.pdf - Назва з екрану. 12.05.2019р.

13. Агрегирование портов. LACP. [Електронний ресурс] : Режим доступу: http://netwild.ru/aggregation/ - Назва з екрану. 05.06.2019р.

14. IEEE 802.1D-2004 - IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges [Електронний ресурс] : Режим доступу: https://standards.ieee.org/standard/802_1D-2004.html - Назва з екрану. 25.04.2019р.

15. Протоколы резервирования МЭК 62439-3 HSR/PRP [Електронний ресурс] : Режим доступу: https://ipc2u.ru/articles/tehnologii-i-innova%D1%81ii/mek-62439-3/ - Назва з екрану. 05.06.2019р.

16. High-Availability Seamless Redundancy (HSR) for IE 4000, IE 4010, and IE 5000 [Електронний ресурс] : Режим доступу: https://www.cisco.com/c/en/us/td/docs/switches/connectedgrid/cg-switch-sw-master/software/configuration/guide/hsr/b_hsr_ie4k.html - Назва з екрану. 15.05.2019р.

17. User Manual DFL-260E/860E/1660/2560/2560G NetDefendOS Version 2.40.03 [Електронний ресурс] : Режим доступу: http://www.dlink.ua/sites/default/files/NetDefendOS_2.40.03_Firewall_UserManual.pdf - Назва з екрану. 08.06.2019р.

18. Understanding EtherChannel Load Balancing and Redundancy on Catalyst Switches [Електронний ресурс] : Режим доступу: https://www.cisco.com/c/en/us/support/docs/lan-switching/etherchannel/12023-4.html - Назва з екрану. 08.06.2019р.

19. Linux Networking: MAC VLANs and Virtual Ethernets [Електронний ресурс] : Режим доступу: http://www.pocketnix.org/posts/Linux%20Networking:%20MAC%20VLANs%20and%20Virtual%20Ethernets - Назва з екрану. 10.06.2019р.

20. Userspace Network interface implementation [Електронний ресурс] : Режим доступу: https://github.com/dmtcp/dmtcp/tree/master/contrib/tun - Назва з екрану. 10.06.2019р.